Frequency Domain Coding of Speech

JOSÉ M. TRIBOLET, MEMBER, IEEE, AND RONALD E. CROCHIERE, SENIOR MEMBER, IEEE

Abstract—Frequency domain techniques for speech coding have recently received considerable attention. The basic concept of these methods is to divide the speech into frequency components by a filter bank (sub-band coding), or by a suitable transform (transform coding), and then encode them using adaptive PCM. Three basic factors are involved in the design of these coders: 1) the type of the filter bank or transform, 2) the choice of bit allocation and noise shaping properties involved in bit allocation, and 3) the control of the step-size of the encoders.

This paper reviews the basic aspects of the design of these three factors for sub-band and transform coders. Concepts of short-time analysis/synthesis are first discussed and used to establish a basic theoretical framework. It is then shown how practical realizations of sub-band and transform coding are interpreted within this framework. Principles of spectral estimation and models of speech production and perception are then discussed and used to illustrate how the "side information" can be most efficiently represented and utilized in the design of the coder (particularly the adaptive transform coder) to control the dynamic bit allocation and quantizer step-sizes. Recent developments and examples of the "vocoder-driven" adaptive transform coder for low bit-rate applications are then presented.

I. Introduction

NEW developments in digital speech communications are evolving at a time when major advances in electronic device technology promise to make implementation practical. This increased capability and decreased cost of digital hardware is prompting an increased interest in more complex and sophisticated coder algorithms which offer better coding quality at lower bit rates. In order to achieve this improved performance, coding techniques must exploit, to a greater degree, information about the mechanisms of speech production and speech perception [44].

Historically, speech coders have been divided into two broad categories, namely, *waveform coders* and *vocoders*. Waveform coders generally attempt to reproduce the original speech waveform according to some fidelity criteria whereas vocoders model the input speech according to a speech production model and then resynthesize the speech from the model. Generally, waveform coders have been more successful at producing good quality, robust speech, whereas vocoders are more fragile and are more dependent on the validity of the speech production model. Vocoders, however, are capable of operating at much lower bit rates (2-5 kbits/s).

In order to reduce the bit rate of waveform coders, recent efforts have focused on taking greater advantage of speech production and speech perception models without making the algorithm totally dependent on these models as in vocoders. A general category of coder algorithms which have been relatively successful in achieving this goal is the class of frequency domain coders. In this class of coders the speech signal is divided into a set of frequency components which are separately encoded. In this way different frequency bands can be preferentially encoded according to perceptual criteria for each band, and quantizing noise can be contained within bands and prevented from creating harmonic distortions outside of the band.

Two basic types of frequency domain coders are considered in this paper, namely, sub-band coders and transform coders. In the first case the speech spectrum is partitioned into a set of, typically, 4-8 contiguous sub-bands by means of a filter bank analysis. In the second case a block by block transform analysis is used to decompose the signal into, typically, 64-512 frequency components. Both techniques, in effect, attempt to perform some type of short-time spectral analysis of the input signal although, clearly, the spectral resolution in the two methods is different. Since both techniques are closely linked to concepts of short-time analysis, these concepts will first be reviewed in Section II. Section III then focuses on a review of concepts of sub-band coding and discusses how they relate to the short-time analysis/synthesis model. Section IV presents a detailed discussion of recent developments and new concepts in "vocoder-driven" adaptive transform coding. Finally, Section V briefly points out other coding techniques which are associated with the class of frequency domain coders.

II. Short-Time Spectral Analysis and Synthesis Framework

The basic concept in frequency domain coding is to divide the speech spectrum into frequency bands or components using either a filter bank or a block transform analysis. After encoding and decoding, these frequency components are used to resynthesize a replica of the input waveform by either filter bank summation or inverse transform means. In this section the basic principles behind these analysis and synthesis methods are discussed. The general framework for this study is provided by the theory of short-time spectral analysis and synthesis. Although practical frequency domain coding schemes, in an effort to tailor themselves to the peculiarities of speech signals, may deviate in one way or another from

Manuscript received January 8, 1979; revised May 14, 1979.

J. M. Tribolet was with the Acoustics Research Department, Bell Laboratories, Murray Hill, NJ 07974. He is now with Instituto Superior Técnico, Lisbon, Portugal.

R. E. Crochiere is with the Acoustics Research Department, Bell Laboratories, Murray Hill, NJ 07974.

0096-3518/79/1000-0512$00.75 © 1979 IEEE

such a framework, it will nonetheless provide important insights into the basic constraints and relationships involved in these coding schemes. This general framework is also invaluable in guiding further research on new methods of frequency domain coding.

A. The Short-Time Fourier Transform

A primary assumption in frequency domain coding is that the signal to be coded is a quasi-stationary (slowly time-varying) signal that can be locally modeled with a short-time spectrum. The objective of frequency domain coding is to isolate the perceptually important components of this short-time spectrum for encoding. Also, for most applications involving real-time constraints, only limited time delays are allowed in the coder and therefore, only a short-time segment of input signal is available at a given time instant.

Within this context a useful definition of a time-dependent short-time Fourier transform is

$$X_n(e^{j\omega}) \triangleq \sum_{m=-\infty}^{\infty} h(n-m)\, x(m)\, e^{-j\omega m} \qquad (1)$$

where $x(m)$ represents samples of the input signal and $h(n-m)$ represents a real "window" which reflects the portion of $x(m)$ to be analyzed. This time dependent transform, known as the *short-time Fourier transform* [1]-[7], is a function of two variables: the discrete time index n, and the continuous frequency ω. It can be interpreted in two convenient ways, either in a filter bank analysis sense or in a block Fourier transform sense. In the filter bank interpretation ω is fixed at $\omega = \omega_0$, and $X_n(e^{j\omega_0})$ is viewed as the output of a linear time-invariant filter with impulse response $h(n)$ excited by the modulated signal $x(n)e^{-j\omega_0 n}$. That is,

$$X_n(e^{j\omega_0}) = h(n) * [x(n)e^{-j\omega_0 n}] \qquad (2)$$

where $*$ denotes the convolution operator. Within this context, $h(n)$ determines the bandwidth of the analysis around the center frequency ω_0 of the signal $x(n)$ and it is referred to as the *analysis filter.*

In the block Fourier transform interpretation the time index n is fixed at $n = n_0$ and $x_{n_0}(e^{j\omega})$ is viewed as the normal Fourier transform of the windowed sequence $h(n_0 - m)$ $x(m)$. That is,

$$X_{n_0}(e^{j\omega}) = F\{h(n_0 - m)\, x(m)\} \qquad (3)$$

where $F\{\ \}$ denotes the Fourier transform. In this context $h(n_0 - m)$ determines the time width of the analysis around the time instant $n = n_0$ and it is referred to as the *analysis window.*

The signal $x(n)$ can be recovered from its short-time spectrum by means of a general synthesis equation or inverse short-time Fourier transform. The following results are based on the general theory of short-time analysis/synthesis developed by Portnoff [1]. The general synthesis equation has the form

$$\hat{x}(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \sum_{r=-\infty}^{\infty} f(n-r)\, X_r(e^{j\omega})\, e^{j\omega n}\, d\omega \qquad (4)$$

where the sequence $f(n)$ is referred to as the *synthesis filter* or the *synthesis window.* By combining (1) and (4), it can be shown that to synthesize and reconstruct $x(n)$ (i.e., $\hat{x}(n) = x(n)$ for all n) an additional relationship must be imposed on the choice of the analysis and synthesis windows, namely that

$$\sum_{n=-\infty}^{\infty} f(-n)\, h(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} F(e^{j\omega})H(e^{j\omega})\, d\omega = 1. \qquad (5)$$

As in the analysis, two particularly convenient interpretations of the short-time Fourier synthesis equation have often been discussed in the literature [3]. The first interpretation occurs when the synthesis window $f(n)$ is chosen to have the form

$$f(n) = \delta(n)/h(0), \qquad h(0) \neq 0. \qquad (6)$$

In this case the synthesis equation (4) becomes

$$\hat{x}(n) = \frac{1}{2\pi h(0)}\int_{-\pi}^{\pi} X_n(e^{j\omega})\, e^{j\omega n}\, d\omega \qquad (7)$$

which can be interpreted as the integral (or incremental sum) of short-time spectral components $X_n(e^{j\omega_0 n})$ modulated back to their center frequencies ω_0. This equation corresponds to a filter bank interpretation of the short-time synthesis.

The second interpretation occurs when the synthesis window is chosen to have the form

$$f(n) = 1/H(e^{j0}) \qquad \text{for all } n. \qquad (8)$$

In this case the general synthesis equation (4) becomes

$$\hat{x}(n) = \frac{1}{H(e^{j0})}\sum_{r=-\infty}^{\infty} F^{-1}\{X_r(e^{j\omega})\} \qquad (9)$$

and it can be interpreted as summing inverse Fourier transformed blocks corresponding to the time signals $h(r-n)\,x(n)$.

As can be seen, there is a direct correspondence between the first interpretations (the filter bank) of the analysis and synthesis methods. Similarly, there is a direct correspondence between the second interpretations (the block-transformation) of the analysis and synthesis methods. It should be noted, however, that these are not the only possible interpretations of short-time spectral analysis and synthesis [1]. In general the synthesis window $f(n)$ is instrumental in exploiting, to a greater or lesser degree, the local correlation of the values of $X_n(e^{j\omega})$ in time n. The two cases discussed above simply correspond to the two extremes of either not exploiting this time correlation at all or exploiting it by giving equal weight to each time instant.

B. The Discrete Short-Time Fourier Transform

An important consideration in the implementation of systems for short-time spectral analysis and synthesis is the choice of sampling rates at which $X_n(e^{j\omega})$ is sampled in both the time and frequency domains. Of special interest to our discussion of frequency domain waveform coding is the problem of formulating this short-time spectral representation with little or no redundancy, where no redundancy implies that there is, on the average, only one sample of the transform representation for each sample of the original signal.

This problem can be formulated in terms of the *discrete short-time Fourier transform.* If $X_n(e^{j\omega})$ is uniformly sampled every R samples in time and every $2\pi/M$ radians in frequency then the discrete short-time Fourier transform, *sampled every R samples in time*, is defined as

$$X_{sR}(k) \triangleq X_{n=sR}(e^{j(2\pi k/M)})$$
$$= \sum_{m=-\infty}^{\infty} h(sR-m)\,x(m)\,e^{-j(2\pi km/M)}. \qquad (10)$$

Similarly, the general synthesis formula has the form

$$\hat{x}(n) = \frac{1}{M}\sum_{k=0}^{M-1}\sum_{s=-\infty}^{\infty} f(n-sR)\,X_{sR}(k)\,e^{j(2\pi kn/M)}. \qquad (11)$$

As in the case of the short-time Fourier transform, two particularly convenient interpretations can be given to the above analysis/synthesis equations, namely, the filter bank and the block transform interpretations. These interpretations will be explored in greater detail in the next sections. Also, as in the short-time Fourier transform, the exact reconstruction of $x(n)$ [i.e., $\hat{x}(n) = x(n)$] implies that the relationship

$$\sum_{s=-\infty}^{\infty} f(n-sR)\,h(pM-(n-sR)) = \delta(p), \qquad \text{for all } n, \qquad (12)$$

must exist between the analysis and synthesis filters. This can also be interpreted in terms of frequency domain constraints [1].

The representation of $x(n)$, without redundancy, in terms of its sampled short-time transform $X_{sR}(k)$ occurs when the decimation period in time R is equal to the number of frequency samples M.

C. Wide-Band Analysis/Synthesis: The Sub-Band Coding Framework

A framework for studying sub-band coding systems can be most conveniently represented by the filter bank interpretation of short-time analysis/synthesis. This interpretation, depicted in Fig. 1, is seen to be that of an M channel filter bank. Analysis consists of modulating the center frequency of each frequency band to dc, low-pass filtering with $h(n)$, and compressing by a factor of $R:1$ [as defined in Fig. 1(a)]. Synthesis consists of expanding [see Fig. 1(b)] the sub-band signal (by filling in with zeros) by a factor of $1:R$, filtering (interpolating) with $f(n)$, and modulating the band center frequency back to its original location. The sub-band signals are then summed to give the output. Although the sub-band coder is rarely implemented in this particular manner, this framework serves as a useful conceptual model for relating various methods of practical implementation, as will be seen later.

Typically, sub-band coders have about 4 to 8 "real" sub-bands or, equivalently, 8 to 16 "complex" bands ($M = 8$ or 16 according to the framework in Fig. 1) [8]-[10]. The bandwidths, therefore, are wide relative to the fine structure (pitch striations) in the voiced speech spectrum, and the sub-band coder can be classified as a wide-band analysis/synthesis system.

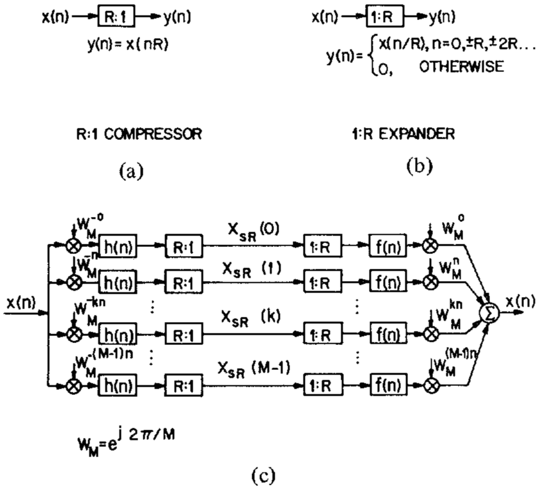


Fig. 1. Filter bank interpretation of short-time analysis/synthesis.

The analysis and synthesis filters $h(n)$ and $f(n)$ in sub-band coders are generally designed to be sharp cutoff low-pass filters with cutoff frequencies of $\pm 2\pi/2M$. In this way sub-bands are isolated as much as possible, avoiding "leakage" of signal energy from one band to another (some amount of "leakage" may be allowed in quadrature mirror filter bands as will be seen later). As a consequence, $X_{sR}(e^{j\omega_k})$ is a low-pass representation of $x(n)$ in the sub-band $\{\omega_k - 2\pi/2M, \omega_k + 2\pi/2M\}$ and it contains relatively little frequency domain aliasing from adjacent sub-bands.

D. Narrow-Band Analysis/Synthesis: The Transform Coding Framework

A framework for studying transform coding systems can be conveniently represented by the block-transform interpretation of short-time Fourier analysis/synthesis. This framework is depicted in Fig. 2. The input signal is divided into time segments which are windowed by the analysis window $h(n)$. Each windowed time segment is transformed to the frequency domain by means of an M point discrete Fourier transform to produce the sampled short-time spectrum $X_{sR}(k)$. Synthesis is achieved by inverse discrete Fourier transforming each sampled short-time spectrum to obtain its (short-time) time domain representation $\hat{x}_{sR}(n)$. The synthesis window $f(n)$ then interpolates across the overlapping short-time signals $\hat{x}_{sR}(n)$ to reconstruct the time signal $\hat{x}(n)$ according to the relation

$$\hat{x}(n) = \sum_{s=-\infty}^{\infty} f(n-sR)\,\hat{x}_{sR}(n) \qquad (13)$$

where $\hat{x}(n) = x(n)$ if the condition in equation (12) is satisfied. Note that, in general, there is no constraint on the width of the synthesis window, i.e., the duration of $f(n)$ can in fact be greater than the transform size M. In this case the inverse transformed short-time signals $\hat{x}_{sR}(n)$ must be interpreted as being periodic in time n with period M.

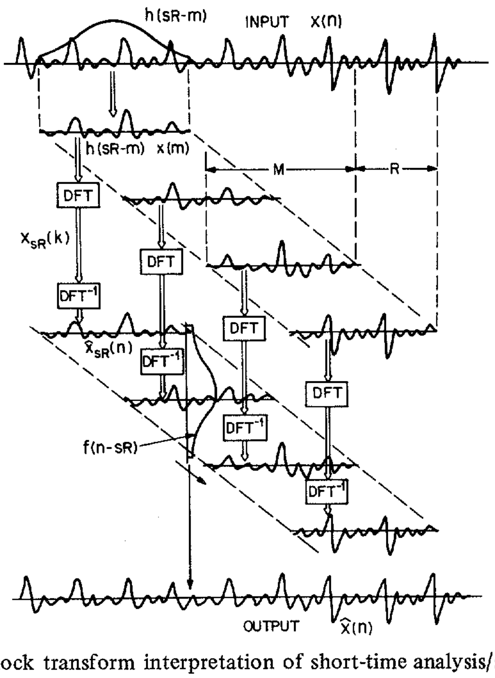


Fig. 2. Block transform interpretation of short-time analysis/synthesis.

In practice, transforms other than the DFT may be employed in transform coding [11]. Of particular interest in this paper, however, are those transforms which have a specific interpretation in terms of the frequency domain. For this class of transforms, this short-time Fourier analysis/synthesis framework will be useful as a conceptual model, as will be seen later.

Typically, the number of frequency channels used in transform coding is much higher than in sub-band coding in order to capitalize on the spectral details (pitch structure) of the speech signal, as well as the general spectral shape (formant structure). Transform sizes on the order of $M = 64$ to 512 have been found to be useful. Thus, transform coding can be classified as a narrow-band analysis/synthesis system. The tradeoff, of course, is that the frequency channels are no longer associated with nonoverlapping frequency bands. Generally, there is a larger amount of "leakage" of signal energy from one band to another.

E. Time and Frequency Aliasing in Analysis/Synthesis

The presence of leakage between frequency bands can affect the performance of a frequency domain coder in two ways. First, if a particular frequency band is low in energy compared to other bands, the energy "leaked" from the other bands can represent a significant portion of the energy in that band. This leakage can interfere with the ability of the coder to take full advantage of the true spectrum of the signal in that band. Secondly, after encoding of the bands, the leakage, or aliasing, from one band to another is not entirely canceled in the synthesis. Therefore, interband leakage in the analysis stage of a frequency domain coder can lead to undesirable effects of frequency domain aliasing in the synthesis stage.

The effects of leakage in frequency bands can be reduced by using analysis and synthesis filters which have lower stopband sidelobes and sharper transition bands than that obtainable with M point filters. This implies that their impulse responses, i.e., the analysis and synthesis windows, must be longer in time. However, if they are longer than the transform size M then the analysis and synthesis time slots overlap in time and aliasing can potentially occur in the time domain as discussed earlier in the block transform interpretation of analysis/synthesis. This time domain aliasing, if it is excessive, can lead to an undesirable reverberant quality in the coder. Thus, in practice, tradeoffs can be made between aliasing effects in time and frequency by changing the size of the analysis and synthesis windows.

As noted earlier, sub-band coding can be characterized as a wideband analysis/synthesis system with sharp cutoff filters to avoid frequency domain aliasing. Therefore, the length of the analysis window in sub-band coding is much longer than its effective transform size. As a result, sub-band coding represents an example where the predominant form of aliasing, due to quantization of the spectral samples $X_{sR}(k)$, is that of time domain aliasing (no aliasing occurs if $X_{sR}(k)$ is not quantized).

Alternatively, transform coding represents the opposite extreme. That is, it is based on a narrow-band analysis with considerable overlap of the analysis filters in the frequency domain. Therefore, the predominant form of aliasing in transform coding, due to quantization of the spectral samples $X_{sR}(k)$, is that of frequency domain aliasing. Again, this aliasing does not occur if the spectral samples $X_{sR}(k)$ are not quantized.

The effects of frequency domain aliasing generally become more pronounced as the dynamic range of the spectrum of the signal being analyzed becomes large. As noted above, one way of controlling this aliasing is by increasing the size of the analysis window (and trading it for time domain aliasing). Another means for controlling frequency domain aliasing is by reducing the dynamic range of the spectrum by preemphasis or spectral flattening prior to the analysis/synthesis. In this way the leakage from large energy bands to low energy bands is reduced. Both fixed and dynamic forms of preemphasis have been widely used in various types of analysis/synthesis speech processing systems for this reason [12], [13]. Fixed preemphasis is generally implemented as a first-order difference filter with an impulse response $W(z) = 1 - \alpha z^{-1}$ where α is on the order of 0.9 for an 8 kHz sampling rate. Dynamic preemphasis is generally accomplished by performing a linear predictive coding analysis on the input signal and then filtering the input signal with the adaptive inverse filter to obtain a spectrally flattened output [12].

F. Discussion

In this section we have briefly reviewed the analysis and synthesis operations involved in sub-band and transform coding and have shown how they can be explained in terms of the unifying framework of short-time spectral analysis/synthesis. These two systems basically differ in the sense that sub-band coders are generally implemented in terms of filter banks whereas transform coders are generally implemented in terms of block transforms.

We wish to point out, however, that when equally spaced frequency bands are considered, each system or interpretation is potentially capable of duplicating the other. In this case, either interpretation can be used in describing sub-band and transform coders. Some difficulties arise, however, when unequally spaced bands are used as, for example, in the sub-band coder.

III. Sub-Band Coding

Sub-band coding has been shown to be an efficient way to exploit the short-time correlations due to the formant structure in speech [8]-[10], [14]. By encoding in sub-bands and allowing the quantizer step sizes in each band to vary independently, the equivalent of a short-time prediction can be achieved. Although this prediction is only obtained in a coarse, piece-wise manner in frequency, it can match the performance of adaptive time domain coding methods with fully adaptive short-time predictors [14]. In very recent work Crochiere and Barabell [42], [43] have demonstrated that pitch structure can also be exploited in sub-band coding. In this section we examine in greater detail the basic principles of the sub-band coder and show how they relate to the filter bank model of wide-band analysis/synthesis. We then review how these techniques can be used for efficient encoding of speech.

A. Filter Bank Implementations

Although the analysis/synthesis filter bank model of Fig. 1 is generally not applied directly in the implementation of sub-band coding, it is closely linked to the interpretation of practical implementations. In practice, sub-bands are generally implemented as a low-pass translation of a frequency band to dc in a manner similar to that of single-side-band modulation. In this way the sub-band signals are real signals as opposed to complex signals as in Fig. 1.

Fig. 3 illustrates the basic frequency domain relationship between these sub-band signals and those in Fig. 1 [8]. The sub-bands with center frequencies at $-\omega_k$ and ω_k and bandwidths B are illustrated in Fig. 3(a). According to Fig. 1, these bands are modulated by the respective signals $e^{j\omega_k n}$ and $e^{-j\omega_k n}$ and low-pass filtered with $h(n)$ to give the resulting complex sub-band signals

$$X_n(e^{\mp j\omega_k}) = a_k(n) \pm jb_k(n)$$
$$= \{x(n) \cos \omega_k n\} * h(n)$$
$$\pm j\{x(n) \sin \omega_k n\} * h(n) \tag{14}$$

where the "+" sign (on the right side of the equation) is associated with the band centered at $-\omega_k$ and the "−" sign is associated with the band centered at $+\omega_k$. The spectra of these sub-band signals are illustrated in Fig. 3(b) and are representative of the sub-band signals in Fig. 1. Fig. 3(c) illustrates a second stage of modulation with the respective signals $e^{-jBn/2}$ and $e^{jBn/2}$ which effectively aligns the upper edge of the band associated with $-\omega_k$ with dc and the lower edge of the band associated with $+\omega_k$ with dc. Summing these two signals then gives the *real* signal $y_k(n)$ as illustrated in Fig. 3(d). This signal can be expressed in the form

$$y_k(n) = 2a_k(n) \cos (Bn/2) + 2b_k(n) \sin (Bn/2) \tag{15}$$

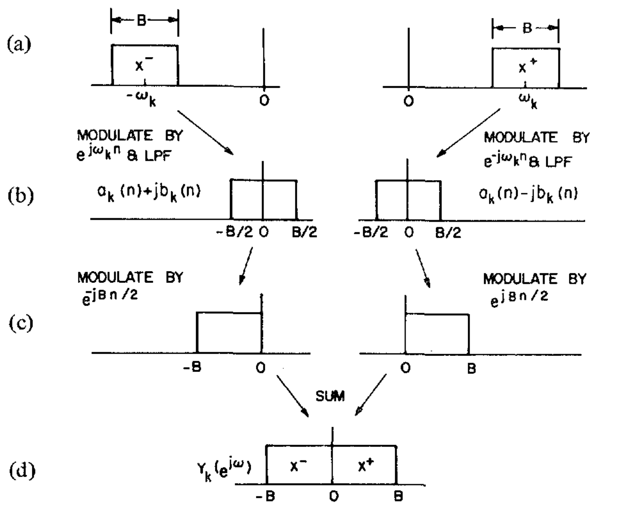


Fig. 3. Frequency domain relationship between sub-band signals and complex filter bank signals.

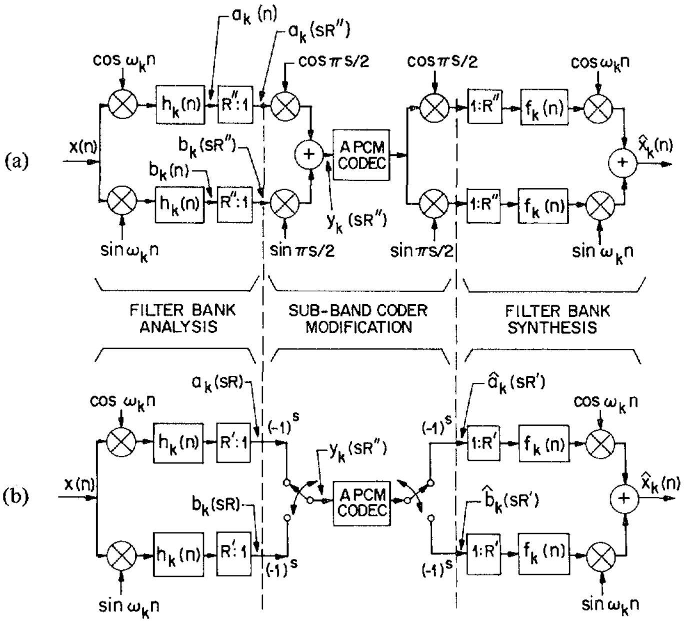


Fig. 4. (a) Block diagram of signal processing operations for sub-band signals. (b) A simplified interpretation showing relationship between $y_k(n)$ and filter bank outputs.

and it is representative of the actual sub-band signal which is generally encoded in sub-band coding.

Fig. 4(a) illustrates this relationship of $y_k(n)$ to $X_n(e^{j\omega_k})$ in terms of a block diagram of signal processing operations. If the bandwidth of $X_n(e^{j\omega_k})$ is $B/2$ radians as illustrated in Fig. 3(b), then the maximum decimation rate of $X_n(e^{j\omega_k})$ is R' where

$$R' = 2\pi/B. \tag{16}$$

However, since $y(n)$ has a bandwidth of B, as seen in Fig. 3(d), the maximum decimation period of $y_k(n)$ is

$$R'' = R'/2 \tag{17}$$

and therefore, this decimation period is also used for $X_n(e^{j\omega_k})$. At this sampling interval, sR'', it can be noted that

$$\cos\left(\frac{B}{2}\cdot sR''\right) = \cos\left(\frac{\pi s}{2}\right) = 1, 0, -1, 0, 1, 0, \cdots \tag{18}$$

and

$$\sin\left(\frac{B}{2}\cdot sR''\right) = \sin\left(\frac{\pi s}{2}\right) = 0, 1, 0, -1, \cdots. \tag{19}$$

Thus, every other sample of the sequences $a_k(sR''/2)$ and $b_k(sR''/2)$ is multiplied by zero and the remaining samples are only changed in their sign. This suggests the interpretation in Fig. 4(b) in which the output of the filter bank analysis is decimated by a factor of $R = R'$ and modulated by $(-1)^s$. The sub-band signal $y_k(n)$ then corresponds to the sequence of interleaved samples of the real and imaginary terms of $X_{sR}(e^{j\omega_k})$, i.e., the outputs of the filter bank model of Fig. 1, with appropriate sign modifications.

Although the block diagram of Fig. 4(b) illustrates a convenient interpretation of how the real output of a sub-band filter for sub-band coding relates to the complex output of an analysis/synthesis filter as in Fig. 1, it does not necessarily suggest the most convenient implementation. In practice, two other methods of implementation are often used. They are the integer-band sampling method (some times referred to as bandpass sampling) [8], [9] and the quadrature mirror filter method [10], [43].

The integer-band sampling method is illustrated in Fig. 5(a). The speech band is filtered by a bandpass filter and the output of the filter is directly decimated by the factor R''. The sub-bands in this implementation are constrained to have lower and upper cutoff frequencies of $K\pi/R'$ and $(K+1)\pi/R'$ where K is an integer associated with the band of interest [see Fig. 5(b)]. The process of decimation by R'' then aliases this band to dc. Similarly, the process of interpolation selects the appropriate Kth "harmonic" of the base band ($K = 0$), thus, effectively bandpass translating it back to its initial frequency. This process of bandpass decimation and interpolation is illustrated in Fig. 5(b) for the case of $K = 2$. An attractive advantage of the integer-band sampling approach is that it eliminates the use of modulators and replaces them with bandpass filters. Therefore, it is efficient in terms of hardware.

The quadrature mirror approach can be developed from a two-band filter bank as shown in Fig. 6(a). This circuit can have two interpretations. The first interpretation is related to that of a two-band version of the analysis/synthesis filter bank in Fig. 1 with the exception that the synthesis filters are not identical from band to band. That is, the synthesis filter for the second band $f_2(n)$ is the negative of that in the first band, i.e.,

$$f_2(n) = -f_1(n) = -f(n) = -h(n)$$
$$n = 0, 1, 2, \cdots \tag{20}$$

where $h(n)$ is assumed to be an *even order* filter [10]. In contrast, the analysis/synthesis formulation in Fig. 1 is based on *odd order* filters.

The second interpretation of the quadrature mirror filter bank can be obtained by combining the $(-1)^n$ modulators with the analysis and synthesis filters in the second band. Thus, a high-pass analysis filter $\tilde{h}_2(n)$ for band 2 can be defined as

$$\tilde{h}_2(n) = (-1)^n h(n)$$
$$n = 0, 1, 2, \cdots \tag{21}$$

and a high-pass synthesis filter $\tilde{f}_2(n) = -\tilde{h}_2(n)$, $n = 0, 1, 2, \cdots$ can be defined which is the negative of the analysis filter. This interpretation has a form similar to the integer-band sampling implementation. In order to avoid gaps between the bands, the additional requirement that

$$1 = |\tilde{H}_1(e^{j\omega})|^2 + |\tilde{H}_2(e^{j\omega})|^2$$
$$= |H(e^{j\omega})|^2 + |H(e^{j(\omega+\pi)})|^2 \tag{22}$$

must be made in the design of the quadrature mirror filter bank [10].

A careful analysis of the quadrature filter bank reveals that, as in the short-time analysis/synthesis formulation of Fig. 1, the frequency domain aliasing terms, i.e., the leakage [illustrated by the shaded region in Fig. 6(b)], cancels down to the level of the quantizing noise in the APCM coders [10]. Therefore, the quadrature mirror filter can be used to trade time domain and frequency domain aliasing effects by adjusting the size of the filter $h(n)$ in a manner similar to the analysis/synthesis structure of Fig. 1.

The quadrature mirror filter bank can be extended to more bands by further subdividing each of the two sub-band outputs with quadrature mirror filters giving a four-band design.

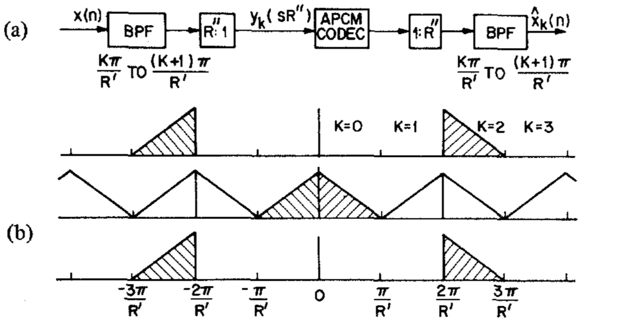


Fig. 5. (a) Integer band sampling. (b) A spectral interpretation.

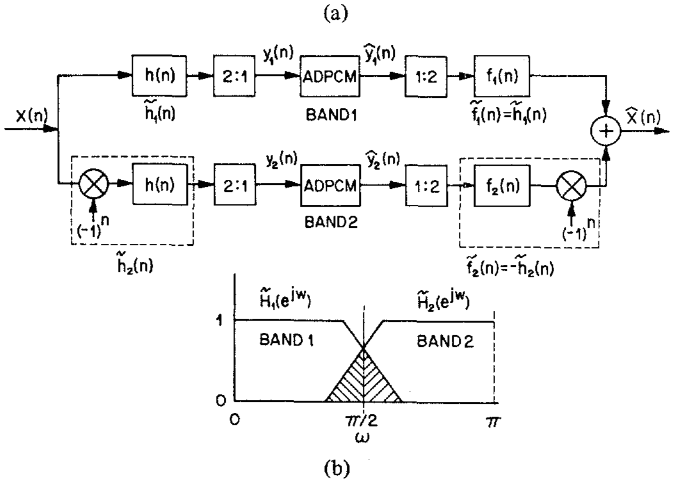


Fig. 6. (a) Quadrature mirror filter bank. (b) A spectral intepretation.

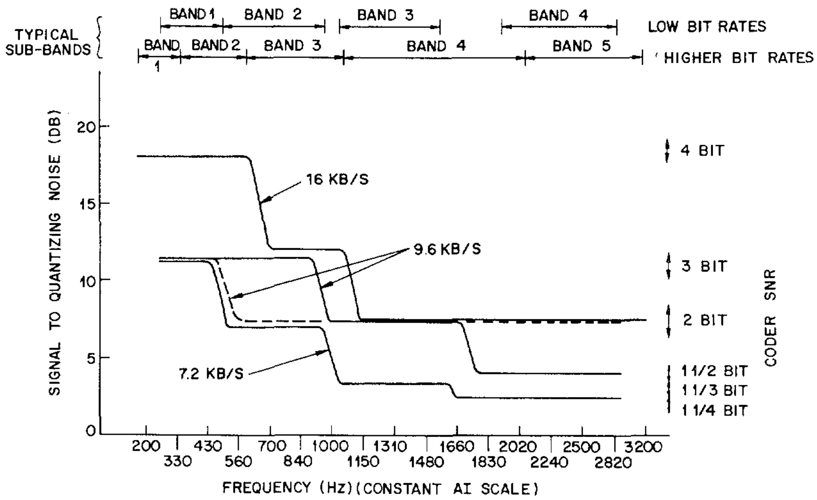


Fig. 7. Choice of sub-bands and bit allocations for sub-band coders with fixed bit allocation.

This "tree structure" can be extended as often as desired to give any number of sub-bands [10].

B. Choice of Sub-Bands in Sub-Band Coding

By dividing the speech band into sub-bands and adaptively encoding each band, sub-band coding is able to take advantage of the "nonflatness" of the speech spectrum due to the formant structure. Since the bandwidth of the formants in speech are typically much narrower (higher Q) at low frequencies than at high frequencies it can be expected that for best performance the sub-bands should have narrower bandwidths at low frequencies and broader bandwidths at high frequencies. An approximate rule of thumb is to choose the bandwidths such that they correspond to significant contributions to the so-called articulation index [8]. Two possible choices for sub-bands are illustrated above Fig. 7 where the frequency scale is nonlinearly warped according to a constant articulation index scale.

C. Bit Allocation and Noise Shaping

The shape of the quantizing noise in frequency can be controlled by the choice of the number of bits/sample used to encode each sub-band. This choice can be made on a fixed basis according to static (long time) perceptual criteria for each sub-band. Alternatively, it can be varied dynamically according to the statistics of the short-time speech spectrum. Although either fixed or dynamic bit allocation can be used in sub-band coding, we will defer the discussion of dynamic bit allocation until the next section on adaptive transform coding.

Typical values of fixed bit allocations for sub-band coders are illustrated in Fig. 7. As seen from this figure, about 12 dB (2 bits) more accuracy is reserved for the lower frequency bands where pitch and formant structure must be more accurately preserved. In upper bands where fricatives and noiselike sounds occur in speech, fewer bits/sample can be used since quantizing noise can be more effectively masked by these sounds.

D. Step-Size Adaptation

The step-sizes of the APCM (adaptive PCM) quantizers are dynamically adjusted to adapt to the speech amplitude in each sub-band. Since this adaptation is performed independently in each band, bands with lower signal energy will have smaller step-sizes and contribute less quantizing noise. Sub-bands with larger signal energy will have larger step-sizes and, therefore, more quantizing noise. This noise, however, will be masked by the larger signal in that band.

The step-size adaptation can be controlled either by a self adapting APCM quantizer or by estimating and transmitting the step-size as additional "side information." The first technique is useful when a fixed bit allocation is used in the coder whereas the second method may be required when a dynamic bit allocation is used. The second method will be described in more detail in the next section on transform coding.

The self adapting step-size for the APCM coders can be based on the one-word memory approach proposed by Jayant, Flanagan, and Cummiskey [15], [16]. The quantizer step-size $\Delta(n)$ is computed according to the relation

$$\Delta(n) = \Delta(n-1) \cdot M(L_{n-1}) \tag{23}$$

where $\Delta(n-1)$ is the step-size at time $n-1$. $M(L_n - 1)$ is a multiplication factor whose value is greater than 1 if an upper quantizer magnitude level L_{n-1} was used at time $n-1$ and less than 1 if a lower quantizer magnitude level was used. In this way the quantizer continuously adapts its step-size in an attempt to track the short-time amplitude level of the speech signal. Other modifications to this algorithm permit improved idle channel performance [17] and robustness to channel errors [18].

E. Discussion

In this section we have attempted to review the basic concepts of sub-band coding and to show how it can be used to take advantage of the formant structure in speech, as well as controlling the shape of the quantizing noise in frequency.

The relationship of the sub-band partitioning and wide-band analysis/synthesis has also been discussed.

Since sub-band coding techniques have been examined in considerable detail in recent literature, specific examples of sub-band coder designs will not be presented here.

IV. Adaptive Transform Coding

In Section II-D it was pointed out that adaptive transform coding can be analyzed in terms of the block transform interpretation of short-time analysis/synthesis. In this section we will examine the principles of transform coding in more detail and show how they relate to this interpretation. We will then report on recent advances in adaptive transform coding, including the new vocoder-driven adaptation strategy [19]. Finally we will present examples of transform coder designs for low bit rate speech communications.

A. Basic Description of Transform Coding

Fig. 8 illustrates a basic block diagram of an adaptive transform coder algorithm as proposed by Zelinski and Noll [11], [40]. The input speech is buffered into short-time blocks of data $x_{sR}(n)$ (as defined in Section II) and transformed. The transformed coefficients, or frequency components, are then adaptively quantized and transmitted to the receiver (as in sub-band coding). At the receiver they are decoded and inverse transformed into blocks $\hat{x}_{sR}(n)$. These blocks are then used to synthesize the output speech signal $\hat{x}(n)$ by a concatenation of the blocks

$$\hat{x}(n) = \sum_{s=-\infty}^{\infty} \hat{x}_{sR}(n). \tag{24}$$

From the discussion of short-time analysis/synthesis in Section II, it can be seen that the above procedure can be interpreted as that of a short-time analysis/synthesis in which the analysis filter $h(n)$ is chosen to be a rectangular window of size M (the transform size) and the decimation period R is also chosen to be $R = M$. The synthesis filter $f(n)$ is $f(n) = 1$ for all n [in accordance with (12)] where the block signals $\hat{x}_{sR}(n)$ are interpreted as being of finite duration M (samples). In the absence of quantization, the output signal $\hat{x}(n)$ is identical to the input $x(n)$.

Although the above analysis/synthesis procedure has been widely used in transform coding [11], [20], [21] it is not clear that it is the most satisfactory for low bit-rate speech coding. More generally, other analysis and synthesis windows may be used which lead to better subjective performance at low bit rates. In Section IV-B we consider this issue and issues concerning the choice of transforms in more detail.

The quantization of the transformed coefficients is assumed to be made with uniform quantizers. The choice of the step-size and the number of bits used for encoding each coefficient is of fundamental importance in transform coding. In the case of stationary inputs, given the input statistics, it is possible to design these quantizers, *a priori*, to meet prescribed specifications for the distribution and minimization of noise in the frequency domain. Speech, however, is a nonstationary signal and this nonstationarity must be properly dealt with. In fact, it has been demonstrated by Zelinski and Noll that transform coding based on long-term characteristics of speech leads to unsatisfactory performance at low bit rates [11].

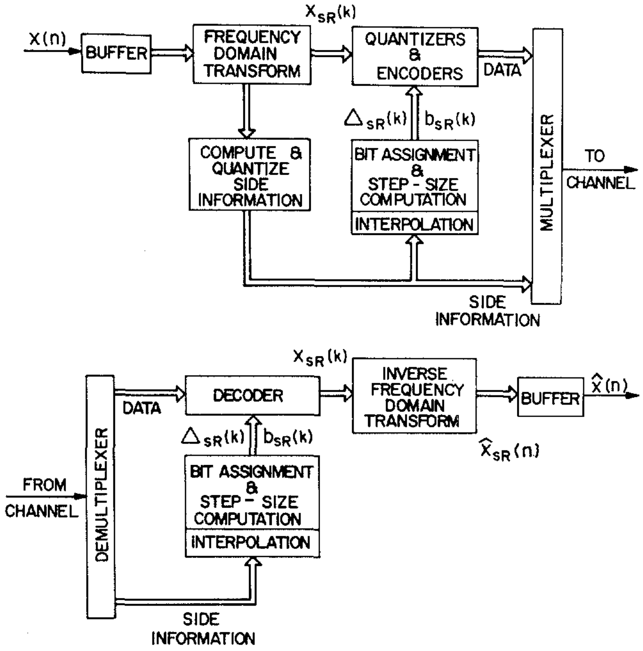


Fig. 8. Block diagram of adaptive transform coder.

To cope with the nonstationarity of speech the step-size $\Delta_{sR}(k)$ in block sR and the number of bits $b_{sR}(k)$ for encoding each transform coefficient k are adaptively changed from block to block. The choice of $\Delta_{sR}(k)$ and $b_{sR}(k)$ is made on the basis of knowledge of the spectral shape of the speech for that block. In sub-band coding it has been seen, in Section III-D, that this knowledge can be acquired by means of a self-adapting step-size algorithm which continually updates its step-size from sample to sample. In transform coding, however, the sampling interval of the coefficients (every R samples) occurs on the order of only once every 16-32 ms which is not sufficient for the self-adapting algorithm. Consequently, the shape and amplitude of the speech spectrum for each block is parameterized, encoded, and transmitted as side information, as seen in Fig. 8. This side information is used in both the transmitter and receiver for step-size adaptation $\Delta_{sR}(k)$ and bit allocation $b_{sR}(k)$.

Four major signal processing operations must therefore be considered in adaptive transform coding. They are the analysis/synthesis operations and the choice of the transform, the spectral parameterization operations, the step-size adaptation and bit allocation (noise shaping) and the quantization and multiplexing of the signals. In the next sections we will consider each of these operations in greater detail.

B. Frequency Domain Transforms and Analysis/Synthesis

In principle, any type of transform can be used in the configuration of Fig. 8. For speech coding, however, there are a number of reasons for restricting this transform to the class of "frequency domain" transforms. First, within the speech coding context the goal is to generate the least *audible* noise possible. Since it is known that the ear makes a short-time frequency analysis of signals [41], it is natural to control the audibility of the quantization by controlling its characteristics in the frequency domain. Secondly, the speech production mechanism can be approximately modeled, on a short-time basis, in terms of linear time-invariant filtering operations. These operations are fairly well understood and provide enormous insight into frequency domain dynamics of speech, thus facilitating the task of adapting the transform coder to the time-varying properties of the speech signal.

Finally, from a purely mathematical point of view, on the basis of a mean-square error criterion, it can be shown that the class of frequency domain transforms asymptotically approach the theoretically optimum performance of the Karhunen-Loeve transforms, in terms of their orthogonalizing properties, for large size transforms [11], [20]-[24]. This theoretical performance can be shown to be related to the ratio of the arithmetic to geometric means of the variance of the short-time speech spectrum [11], [20], [23]. Although the mean-square error criterion is not necessarily the most appropriate criterion in terms of speech perception, it is satisfying to note that these results are in general agreement with the above physical arguments for using frequency domain transforms.

Zelinski and Noll have examined, in great detail, the properties of a number of transforms for speech coding purposes, including two frequency domain transforms, the discrete Fourier transform (DFT) and the discrete cosine transform (DCT) [4]. Their results experimentally verify the asymptotic optimality of the frequency domain transforms. Furthermore, they have demonstrated that for speech signals the DCT is nearly optimal in terms of its performance compared with the Karhunen-Loeve transform (a result also found in image coding [24]). In comparison to the DFT, they found that the DCT has about 4-5 dB better S/N performance, for many speech sounds, with transform sizes of $M = 128$ (although both transforms are asymptotically optimal as M becomes very large). Since the Karhunen-Loeve transform is a data-dependent transform and the DCT is a fixed transform, the DCT is generally preferable in terms of a practical implementation.

Formally, the DCT of a real M point sequence $v(n)$ can be defined as

$$V_c(k) = \sum_{n=0}^{M-1} v(n)\, c(k) \cos\left[(2n+1)\pi k/2M\right]$$
$$k = 0, 1, 2, \cdots, M-1 \qquad (25)$$

where

$$c(k) = \begin{cases} 1 & k = 0 \\ \sqrt{2} & k = 1, 2, \cdots, M-1. \end{cases} \qquad (26)$$

Similarly, the inverse DCT can be defined as

$$v(n) = \frac{1}{M} \sum_{k=0}^{M-1} V_c(k)\, c(k) \cos\left[(2n+1)\pi k/2M\right]$$
$$n = 0, 1, 2, \cdots, M-1. \qquad (27)$$

It can be seen from (25) that the DCT coefficients $V_c(k)$ are real numbers for real $v(n)$ and they correspond, respectively, to the M frequencies $\omega_k = 2\pi k/2M$, $k = 0, 1, \cdots, M-1$, which are equally spaced around the upper half of the unit circle.

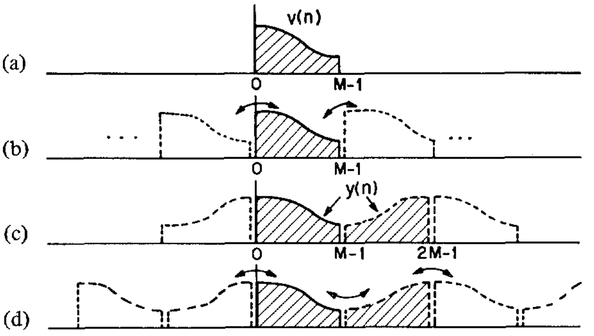


Fig. 9. (a) One block of data $v(n)$. (b) Illustration of end effects for DFT analysis/synthesis. (c) Equivalent $2M$ point data block $y(n)$ for DCT analysis. (d) Illustration of end effects for DCT analysis/synthesis.

The near optimal performance of the DCT has been attributed in recent literature to the fact that the basis vectors of the DCT closely approximate the eigenvectors of a class of Toeplitz matrices [24]. In this paper, however, in an effort to relate the DCT to concepts of short-time analysis/synthesis, we will present an additional reason, based on digital signal processing concepts, as to why the DCT is preferable.

Consider first a short-time analysis/synthesis based on the DFT. Let $v(n)$ $n = 0, 1, \cdots, M-1$ be one block of data of length M as depicted in Fig. 9(a) and $V(k)$ $k = 0, 1, \cdots, M-1$ be its transform. If $V(k)$ is quantized with a relatively large number of bits for each coefficient, then the quantizing noise can be modeled as an additive noise and the quantized version $\hat{V}(k)$ can be represented as

$$\hat{V}(k) = V(k) + E_v(k) \qquad (28)$$

where $E_v(k)$ represents the noise component. The inverse transformation, in the synthesis, leads to the signal $v(n) + e_v(n)$, which is simply the original signal plus an additive noise in time. If, however, the total number of bits used to encode $V(k)$ is very low, as in low bit-rate coding, then the quantizing noise is a combination of both multiplicative and additive effects, i.e.,

$$\hat{V}(k) = G_v(k)\, V(k) + E_v(k) \qquad (29)$$

[where $E_v(k)$ is no longer the same as in (28)]. In fact, some values of $V(k)$ may not be encoded at all in which case $V(k) = 0$ for those coefficients. For example, if the high-frequency DFT components have very low energy, as in typical voiced regions, the entire upper frequency range may not be encoded, leading to a low-pass effect. The synthesis procedure then leads to the result

$$\hat{v}(n) = v(n) \otimes g_v(n) + e_v(n) \qquad (30)$$

where $g_v(n)$ is the inverse transform of $G_v(k)$ and $\otimes$ denotes the *circular* convolution of $v(n)$ with $g_v(n)$. This circular convolution results in an exchange of energy between the left and right boundaries of $v(n)$ (i.e., aliasing in time), as illustrated by the arrows in Fig. 9(b). These end effects can lead to very undesirable "click" and "burbling" noises at the block rate in transform coding.

A well-known solution to the above aliasing problem is to pad $v(n)$ with zeros (equal to the number of samples of $g_v(n)$ minus 1) and use a larger transform. Unfortunately, for transform coding, $g_v(n)$ and $e_v(n)$ are not well-defined time-limited quantities, and increasing the transform size to be larger than the block size only reduces the average number of bits/sample across the block which further compounds the problem.

The DCT (and a number of closely related transforms) reduces the above end-effect problems between the left and right boundaries while still keeping a minimum spectral redundancy (i.e., M transform coefficients for M data points). As pointed out by Chen and Fralick [25], the DCT is closely related to the $2M$ point DFT of a sequence $y(n)$, which is formed from the M point sequence $v(n)$, by defining

$$y(n) = \begin{cases} \frac{1}{2}\, v(n) & n = 0, 1, \cdots, M-1 \\ \frac{1}{2}\, v(2M-1-n) & n = M, M+1, \cdots, 2M-1. \end{cases} \tag{31}$$

The sequence $y(n)$ is depicted by the shaded region in Fig. 9(c). The $2M$ point DFT of $y(n)$ leads to

$$Y(k) = \sum_{n=0}^{2M-1} y(n)\, e^{-j(2\pi kn/2M)} \tag{32}$$

$$= e^{j(\pi k/2M)} \sum_{n=0}^{M-1} v(n) \cos\,[(2n+1)\pi k/2M]. \tag{33}$$

By comparison of (33) with (25), it can then be seen that the DCT of $v(n)$ can be obtained from $Y(k)$ according to the relation

$$V_c(k) = c(k)\, e^{-j\pi k/2M}\, Y(k) \qquad k = 0, 1, 2, \cdots, M-1. \tag{34}$$

While more efficient methods for computing the DCT are available than that of performing a $2M$ point DFT [26], [27], the above interpretation is particularly useful for gaining insight into the properties of the DCT. Because of the association of the DCT with the $2M$ point DFT of the symmetric sequence $y(n)$ (symmetric about a "half sample"), it can be seen that quantizing $V_c(k)$ is, in effect, equivalent to quantizing $Y(k)$. Therefore, as in the DFT analysis, we can write

$$\hat{Y}(k) = G_y(k)\, Y(k) + E_y(k) \tag{35}$$

and

$$\hat{y}(n) = g_y(n) \otimes y(n) + e_y(n). \tag{36}$$

Fig. 9(d) depicts the end effects between the left and right boundaries of $\hat{y}(n)$ due to the circular convolution of $y(n)$ with the multiplicative component of quantization $g_y(n)$. In terms of the sequence $v(n)$, however, it is seen that these interactions are now localized, and there is no longer an exchange of energy between the left and right boundaries of $v(n)$. Thus, the DCT produces less noticeable boundary effects than the DFT in transform coding applications, a result which is in good agreement with observations found in the literature [11], [24], [26]. This is an additional advantage of the DCT besides the fact that it is "close" in performance to the Karhumen-Loeve transform.

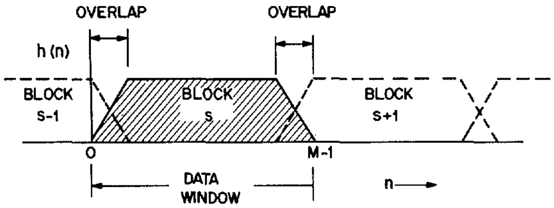


Fig. 10. Trapezoidal window for DCT analysis/synthesis.

The choice of an analysis window $h(n)$ in the analysis/synthesis involving the DCT can be instrumental in further reducing the boundary effects. Fig. 10 illustrates a class of trapezoidal windows that have been found to be very useful for low bit-rate coding. By allowing a small (10 percent or less) overlap between the successive blocks being coded, a significant reduction of end-effect noise can be achieved without significantly lowering the number of bits available for encoding each block. Clearly, if the number of bits is sufficiently reduced, the overall increase in quantization noise can offset whatever noise reduction is achieved by the overlapping process.

C. A Spectral Interpretation of the DCT

An alternate way of expressing an M point DCT in terms of a $2M$ point DFT was initially proposed by Ahmed and Rao [24]. Using this interpretation and the concepts of short-time analysis/synthesis an interesting spectral interpretation of the DCT can be derived. In this section this interpretation will be discussed and the results will be utilized in later sections.

Let $u(n)$ denote an M point sequence such that $u(n) = v(n)$ for $0 \leqslant n \leqslant M-1$ and $u(n) = 0$ elsewhere. Then the $2M$ point DFT of $u(n)$ is

$$U(k) = \sum_{n=0}^{2M-1} u(n)\, e^{-j(2\pi kn/2M)} \tag{37a}$$

$$= \sum_{n=0}^{M-1} u(n)\, e^{-j(2\pi kn/2M)} \tag{37b}$$

$$= e^{j(k\pi/2M)} \sum_{n=0}^{M-1} u(n)\, e^{-j(\pi k(2n+1)/2M)}$$

$$k = 0, 1, 2, \cdots, 2M-1. \tag{37c}$$

From (25) it then becomes clear that the M point DCT of $v(n)$, denoted $V_c(k)$, can be expressed in terms of $U(k)$ according to

$$V_c(k) = R_e\, \{c(k)\, e^{-j(\pi k/2M)}\, U(k)\} \qquad k = 0, 1, 2, \cdots M-1. \tag{38}$$

where R_e denotes the real part. Denoting $|U(k)|$ and θ_k as the magnitude and phase of $U(k)$, (38) can be expressed in the form

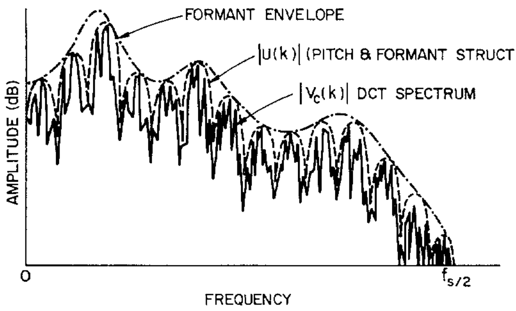


Fig. 11. Illustrative interpretation of DCT spectrum.

$$V_c(k) = R_e \; \{c(k) \, | U(k) | \, e^{j(\theta_k - (\pi k/2M))}\} \tag{39}$$

$$V_c(k) = c(k) \, | U(k) | \cos(\theta_k - \pi k/2M) \qquad k = 0, 1, 2, \cdots M - 1. \tag{40}$$

Thus, it is seen that the DCT has a spectral envelope which is identical to that of the DFT and a modulating term, $\cos(\theta_k - \pi k/2M)$, which adds a rapidly varying component to its spectrum. Fig. 11 gives an illustrative example of this DCT spectrum (this is an illustration only, it is not obtained from real speech). Since the DCT is bounded by the spectral envelope of the DFT, it is also apparent that it exhibits all of the properties of formant structure and pitch striations that are present in the DFT spectrum. These speech characteristics can therefore be exploited directly with the short-time analysis/synthesis based on the DCT.

By appropriately defining the analysis window $h(n)$ in short-time analysis/synthesis, an equivalent filter bank model for the DCT, based on a $2M$ channel filter bank (with M redundant channels), can be described. Let

$$\omega_k = \pi k/M \qquad k = 0, 1, \cdots, 2M - 1 \tag{41}$$

denote the center frequencies of the $2M$ channels of the filter bank. Then, from (10) and (38), an appropriate definition of the short-time DCT, sampled at times $n = sR$, can be given as

$$X_{cn}(k)\,|_{n=sR} = R_e \left\{ c(k) \, e^{-j\omega_k/2} \sum_{m=-\infty}^{\infty} h(n-m) \cdot x(m) \, e^{-j\omega_k m} \right\} \tag{42a}$$

$$= R_e \; \{c(k) \, e^{-j\omega_k/2} \, [(x(n) \, e^{-j\omega_k n}) * h(n)] \}$$

$$= c(k) \cos(\omega_k/2) [(x(n) \cos(\omega_k n)) * h(n)] \tag{42b}$$

$$- c(k) \sin(\omega_k/2) [(x(n) \sin(\omega_k n)) * h(n)]. \tag{42c}$$

The above equations represent the filter-bank interpretation of the DCT as shown in Fig. 12 (for one analysis channel). The model can be divided into two parts, the first part which consists of a $2M$ channel DFT filter bank as in Fig. 1 and the second part which consists of the modification due to the DCT. The close association of the DCT analysis to the short-time Fourier analysis is therefore readily apparent from this model and Fig. 11.

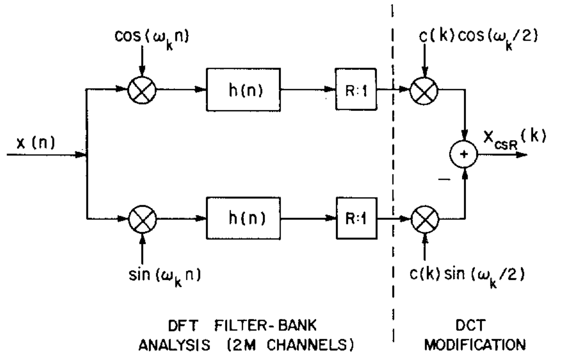


Fig. 12. Filter bank analysis model for DCT.

D. Quantization of the Transform Coefficients

The quantization of the transform coefficients is usually made by means of uniform quantizers which are characterized by a step-size $\Delta_{sR}(k)$ and by a number of levels $2^{b_{sR}(k)}$. The choice of the step-size and the number of bits $b_{sR}(k)$ for a given transform coefficient is of fundamental importance in adaptive transform coding. In this section we assume that the bit allocation has already been determined and that an estimate of the spectral variance $\hat{\sigma}_{sR}^2(k)$ of the transform coefficients is known. The bit allocation and estimation of spectral variance will be discussed in greater detail in Sections IV-E and IV-F.

As observed by Zelinski and Noll [11], the probability density functions of the (gain normalized) transform coefficients are approximately Gaussian distributed. Therefore, the choice of the optimum (uniform) step-size $\Delta_{sR}(k)$, considering a mean-square error criterion, can be determined from the variance estimate $\hat{\sigma}_{sR}^2(k)$ according to the theory of Max [28]. For a given number of bits $b_{sR}(k)$, the optimum step-size is therefore

$$\Delta_{sR}(k) = \alpha(b_{sR}(k)) \; \hat{\sigma}_{sR}(k) \tag{43}$$

where $\alpha(b_{sR}(k))$ is a constant of proportionality, which is a function of the number of bits, and can be found in the tables of Max.

From the point of view of subjective quality, however, it is not clear that a mean-square error criterion is the most appropriate choice for determining the step-size. Therefore, in practice, we found that it is desirable to include an additional factor Q, denoted as the quantizer loading factor, in the equation of (43). Thus,

$$\Delta_{sR}(k) = Q\alpha(b_{sR}(k)) \; \hat{\sigma}_{sR}(k) \tag{44}$$

where $Q = 1$ implies a loading that is optimum in the mean-square (uniform step-size) sense. By adjusting Q, a trade can be made between effects of overload and granular types of distortion in the transform coder. The effect of Q on the subjective quality of the coder will be discussed in greater detail in Section IV-G.

E. Bit Allocation and Noise Shaping

The choice of the bit allocation $b_{sR}(k)$ determines the accuracy in which the transform coefficients are encoded. Thus it controls the distribution of the quantizing noise in the frequency domain. An extensively studied case is that of a stationary Gaussian correlated random process [20], [23]. If the transform coefficients have variances $\sigma_{sR}^2(k)$, and if a minimum mean-square error criterion is desired, then the optimum bit assignment $b_{sR}(k)$ can be shown to be

$$b_{sR}(k) = \delta + \frac{1}{2}\log_2 \frac{\sigma_{sR}^2(k)}{D^*} \qquad k = 0, 1, 2, \cdots, M-1 \quad (45)$$

where δ is a correction term that takes into account the performance of practical quantizers. D^* denotes the variance of the quantization noise and is defined as

$$D^* = \frac{1}{M}\sum_{k=0}^{M-1} \sigma_e^2(k) \quad (46)$$

where $\sigma_e^2(k)$ denotes the variance of the quantization noise incurred in quantizing the kth transform coefficient. The value of D^* is chosen such that the sum of the bit assignments $b_{sR}(k)$ satisfies the constraint

$$B = \sum_{k=0}^{M-1} b_{sR}(k) \quad (47)$$

where B is the number of bits/block available for transmission over the binary channel. It can be shown also that the above bit assignment rule, based on a minimum mean-square error over the block, leads to a flat noise distribution in frequency [11], [20], [23], [40], i.e., $\sigma_e^2(k) = D^*$ for all k.

An interpretation of this bit assignment rule can be seen in Fig. 13. The dashed horizontal lines represent decision thresholds, λ_i, for choosing the bit allocation $b_{sR}(k)$. For example, if the kth log spectral coefficient $\log_2 \sigma_{sR}^2(k)$ lies between λ_3 and λ_4, the bit allocation for that coefficient is $b_{sR}(k) = 4$. The thresholds are spaced 6 dB apart. Thus, for every 6 dB that $\sigma_{sR}^2(k)$ is increased, one more bit, or 6 dB of signal-to-noise-ratio, is added to the quantizer. Therefore, the noise remains flat across the frequency band. Two exceptions to this rule occur. All values of $\log_2 \sigma_{sR}^2(k)$ below λ_0 are assigned zero bits (negative bits are not allowed) and all values of $\log_2 \sigma_{sR}^2(k)$ above, say λ_5, are assigned 5 bits (i.e., there is a maximum limit). If the resulting total number of bits assigned in this way is less than the number of bits available for transmission B then the level of the thresholds λ_i (proportional to D^*) are uniformly reduced (keeping a 6 dB spacing). If the total number of bits is greater than B, the threshold level is increased. This process continues until (47) is satisfied. In practice, this bit allocation can be achieved by a three-step process [39] instead of an iterative process as implied above.

As observed, the above bit allocation scheme results in a quantization noise $\log_2 \sigma_e^2(k)$ that is flat across the spectrum and is proportional to the threshold level λ_0. In terms of a mean-square error criterion it can be shown that this algorithm minimizes the noise variance $\sigma_e^2(k) = D^*$. From perceptual criteria, however, it is known that a flat noise distribution is not the most desirable. To take into account the shape of the quantization noise in frequency, the bit assignment rule of (45) can be modified by allowing a (positive) weighting factor $w(k)$ that weights the importance of the noise in different frequency bands. Thus, (45) becomes

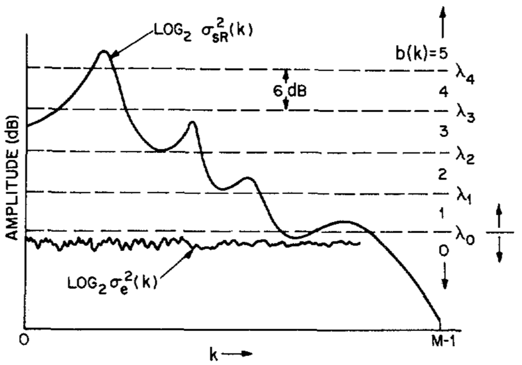


Fig. 13. Interpretation of bit assignment rule.

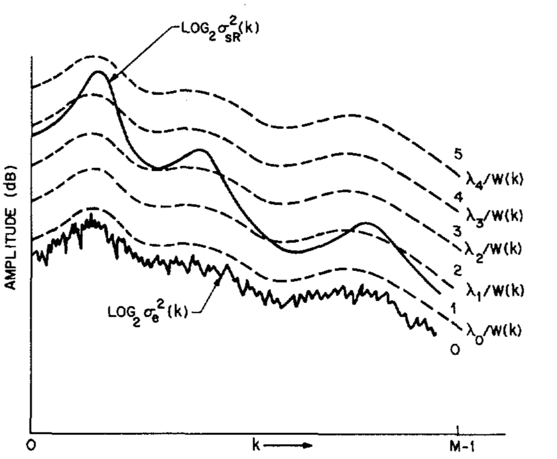


Fig. 14. Interpretation of frequency-weighted bit assignment.

$$b_{sR}(k) = \delta + \frac{1}{2}\log_2 \frac{w(k)\,\sigma_{sR}^2(k)}{D^*} \qquad k = 0, 1, 2, \cdots, M-1. \quad (48)$$

This bit assignment minimizes the following frequency weighted distortion measure:

$$D^* = \frac{1}{M}\sum_{k=0}^{M-1} w(k)\,\sigma_e^2(k). \quad (49)$$

The resulting noise spectrum is then given by

$$\sigma_e^2(k) = L \cdot (w(k))^{-1} \cdot D^* \qquad k = 0, 1, 2, \cdots, M-1 \quad (50)$$

where L is a constant. One interpretation of this frequency weighted bit assignment is depicted in Fig. 14 where it is seen that the thresholds λ_i are modified by $w(k)$. Alternatively, it can be viewed as a preemphasized $\sigma_{sR}^2(k)$, i.e., $w(k)\,\sigma_{sR}^2(k)$, and flat thresholds in a manner similar to that in Fig. 13.

The question remains as to what the most appropriate form of weighting $w(k)$ is for optimum subjective performance of the transform coder. In general, it can be observed that this weighting should be a dynamic one, e.g., the most appropriate weighting for voiced speech will be different than that for unvoiced speech. More specifically, the weighting should be chosen in a manner such that the quantization noise is most effectively masked by the speech signal [29]-[31], [44].

One class of weighting functions that provides a wide range of control over the shape of the quantizing noise relative to the shape of the speech spectrum is given by the functional form

$$w_{sR}(k) = \sigma_{sR}^{2\gamma}(k) \qquad k = 0, 1, 2, \cdots, M-1 \quad (51)$$

where γ is a parameter that can be experimentally varied. The case where $\gamma = 0$ (uniform weighting) has been discussed previously. The noise spectrum in this case is flat and the bit assignment is such that the signal-to-noise-ratio has the shape of the spectrum. The case where $\gamma = -1$ (inverse spectral weighting) leads to a constant bit assignment. Here the noise spectrum will follow the input spectrum, and the signal-to-noise-ratio is constant as a function of frequency.

As the value of γ is slowly varied between these two extremes $(-1 < \gamma < 0)$, the noise spectrum will likewise evolve from a flat distribution to one that precisely follows that of the speech spectrum. This variation is depicted in Fig. 15. Note that the spectral estimate in this illustration does not include details about the fine structure (pitch harmonics) in the spectrum. When the pitch structure is considered, the form of the weighting in (51) should be modified such that it follows only the smooth (formant) component of the spectral model. This will be discussed in more detail in Section IV-F. The choice of appropriate values for γ will be discussed in more detail in Section IV-F.

F. Spectral Parameterization and Adaptation of the Transform Coder

The application of the above bit assignment and step-size adaptation algorithms are strongly dependent on the estimate of the spectral variance $\sigma_{sR}^2(k)$. The more accurate this estimate is of the true variance the better the performance and the more reliable the above algorithms will be. Since speech is a nonstationary process these spectral variances are not known *a priori* and must therefore be estimated, encoded, and transmitted to the receiver. This information, which represents in some form the dynamical properties of speech in the transform domain, is commonly referred to as "side information."

Two basic adaptation techniques for transform coding of speech have been proposed in recent literature. The first technique, proposed by Zelinski and Noll [11], [40] is illustrated in Fig. 16. The DCT spectrum is represented by a reduced set of (typically 16 to 24) equally spaced samples of the spectral estimate. These samples are computed by a local averaging of the DCT magnitude coefficients around the sample frequencies. The sample values are quantized and encoded for transmission to the receiver as side information (as seen in Fig. 8). They are also decoded and used in the transmitter so that the step-size and bit allocation computations are exactly duplicated in the transmitter and the receiver. The encoding of the side information requires approximately 2 kbits/s. Further details on a modification of this encoding procedure are given in Section IV-G.

To obtain spectral estimates of $\sigma_{sR}(k)$ at all frequencies (i.e., all values of k), the above sample estimates are geometrically interpolated (i.e., linearly interpolated in the log domain), as illustrated in Fig. 16. The result is a piecewise approximation of the spectral levels in the frequency domain. These values of $\hat{\sigma}_{sR}(k)$ are then used by the bit assignment and step-size adaptation algorithms.

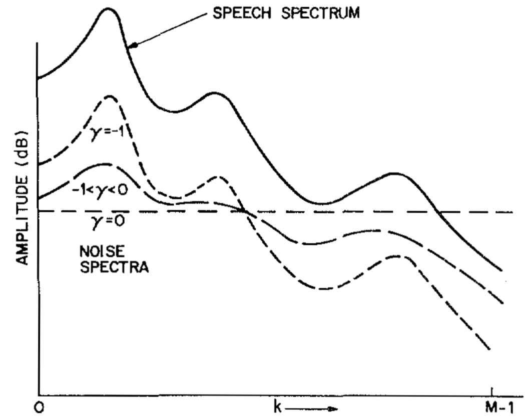


Fig. 15. Control of noise shaping by parameter γ.

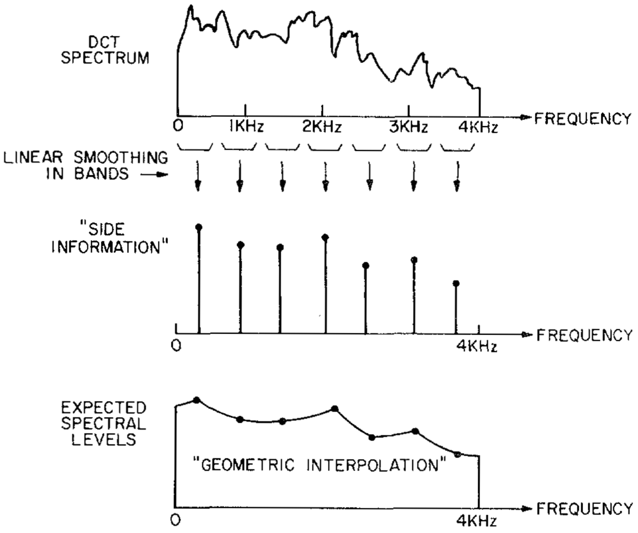


Fig. 16. Representation of side information as equal spaced samples of the spectral estimate.

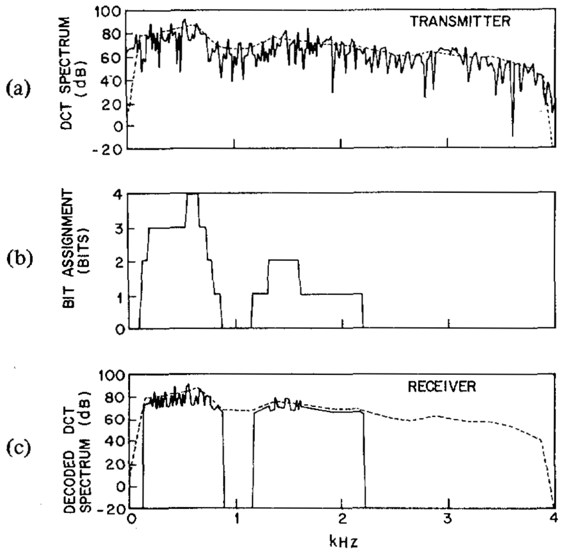
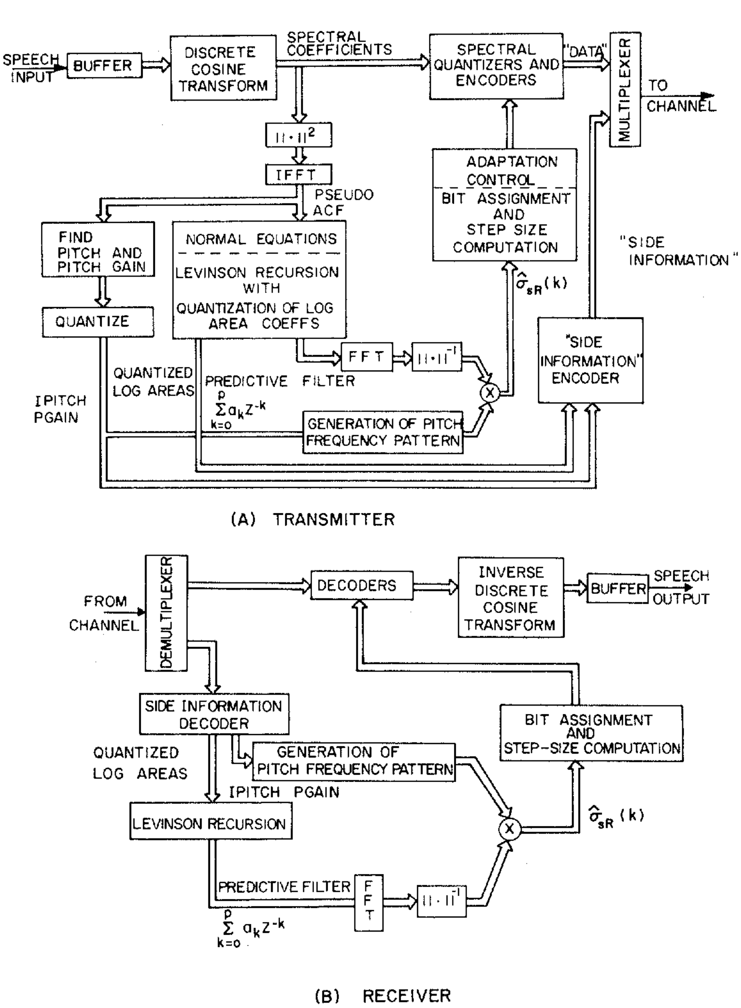


Fig. 17. Illustration of the operation of the adaptation scheme of Fig. 16.

Fig. 17 illustrates the operation of the above adaptation scheme for transform coding at 8 kbits/s. Fig. 17(a) shows the DCT spectrum and the estimated spectral levels $\hat{\sigma}_{sR}(k)$ as seen by the dotted line. Fig. 17(b) shows the resulting bit assign-


SPEECH INPUT
BUFFER
DISCRETE COSINE TRANSFORM
SPECTRAL COEFFICIENTS
SPECTRAL QUANTIZERS AND ENCODERS
"DATA"
MULTIPLEXER
TO CHANNEL
$|\cdot|^2$
IFFT
PSEUDO ACF
ADAPTATION CONTROL
BIT ASSIGNMENT AND STEP SIZE COMPUTATION
"SIDE INFORMATION"
FIND PITCH AND PITCH GAIN
NORMAL EQUATIONS
LEVINSON RECURSION WITH QUANTIZATION OF LOG AREA COEFFS
QUANTIZE
$\hat{\sigma}_{sR}(k)$
FFT
$|\cdot|^{-1}$
"SIDE INFORMATION" ENCODER
QUANTIZED LOG AREAS
PREDICTIVE FILTER $\sum_{k=0}^{p} a_k z^{-k}$
IPITCH PGAIN
GENERATION OF PITCH FREQUENCY PATTERN
FROM CHANNEL
DEMULTIPLEXER
DECODERS
INVERSE DISCRETE COSINE TRANSFORM
BUFFER
SPEECH OUTPUT
SIDE INFORMATION DECODER
BIT ASSIGNMENT AND STEP-SIZE COMPUTATION
QUANTIZED LOG AREAS
GENERATION OF PITCH FREQUENCY PATTERN
IPITCH PGAIN
LEVINSON RECURSION
$\hat{\sigma}_{sR}(k)$
PREDICTIVE FILTER
FFT
$|\cdot|^{-1}$
$\sum_{k=0}^{p} a_k z^{-k}$

(A) TRANSMITTER

(B) RECEIVER

Fig. 18. Block diagram of "spech specific" or "vocoder-driven" adaptation algorithm for ATC.

ment obtained with this spectral estimate and Fig. 17(c) shows the decoded DCT spectrum at the receiver. Because of the low bit rate (8 kbits/s), large regions of the spectrum receive essentially no bits for encoding. Also, in regions where only one bit is used for encoding, this bit must be used for the sign, and therefore the quantized magnitude is proportional to $\hat{\sigma}_{sR}(k)$. In these regions, as for example, the region near 2 kHz in Fig. 17(c), it is seen that all information concerning the spectral detail is lost.

We refer to the above adaption algorithm as a "nonspeech specific" algorithm in the sense that it does not directly take into account the known properties of speech, such as the all-pole vocal-tract model and the pitch model. The technique, however, is quite appropriate for speech transmission at or above 16 kbits/s, since at such rates there are sufficient bits to allow an accurate representation of the fine structure (pitch harmonics) in the DCT spectrum. As the bit rate is reduced below 16 kbits/s, however, it becomes increasingly more difficult to accurately encode the fine structure. In fact, at 8 kbits/s, for example, the pitch information is no longer sufficiently preserved and, as a consequence, the received signal appears degraded by a very perceptible "burbling" or "click" distortion.

One way of making the above algorithm slightly more tailored to speech is to use an unequal spacing for the sampled estimates [40]. One criterion is to use an articulation based scale such that sampled estimates are more closely spaced at lower frequencies and more widely spaced at higher frequencies. The reasoning is similar to that for choosing unequally spaced bands for the sub-band coder. At low frequencies the Q's of the formant resonances are generally much higher than at high frequencies. Thus, the spectrum typically varies more at low frequencies than at high frequencies. While this modification improves the performance of ATC slightly, it is not sufficient to overcome the difficulties mentioned above at low bit rates (below 16 kbits/s).

A more appropriate algorithm for bit rates below 16 kbits/s is a more complex "speech specific" adaptation algorithm which takes full advantage of the known models and dynamics of the speech production mechanism in order to predict the DCT spectral levels [19]. This algorithm is based on an all-pole model of the formant structure of speech and a pitch model to represent the fine structure (pitch striations) in the speech spectrum [12], [13]. The resulting algorithm is referred to as a "vocoder-driven" adaptation strategy due to the close relationship of this spectral estimate to a vocoder model.

Fig. 18 illustrates a block diagram for one possible implementation of this technique. First the DCT spectrum is squared and inverse transformed with an inverse DFT. This yields an autocorrelation-like function which we shall refer to

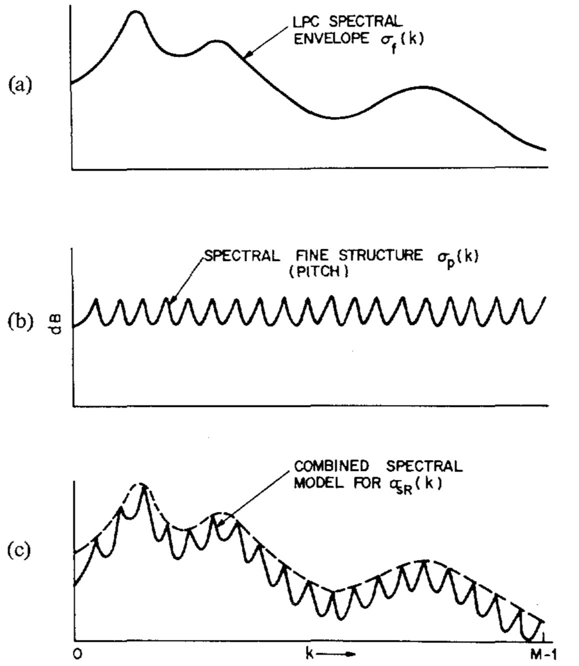

(a)
LPC SPECTRAL ENVELOPE $\sigma_f(k)$
(b)
dB
SPECTRAL FINE STRUCTURE $\sigma_p(k)$ (PITCH)
(c)
COMBINED SPECTRAL MODEL FOR $\sigma_{SR}(k)$
0
k
M-1

Fig. 19. Spectral components of speech spectrum model. (a) Formant structure. (b) Pitch structure. (c) Combined model.

as the pseudo-ACF (autocorrelation function). Since the DCT spectrum is bounded in shape by the Fourier spectrum as seen in Section IV-C, this pseudo-ACF exhibits very similar properties to that of a normal ACF. The first $P + 1$ values of this function are used to define a correlation matrix in the usual normal equations formulation sense [12]. The solution of these equations yields an LPC filter of order P. The inverse spectrum, illustrated in Fig. 19(a), yields an estimate of the formant structure of the DCT spectrum denoted as $\sigma_f(k)$.

The fine structure of the DCT spectrum is obtained from a pitch model. To obtain the pitch period l the pseudo-ACF is searched for a maximum above the range $P + 1$. The corresponding pitch gain G is the ratio of the pseudo-ACF at l over its value at the origin. With these two parameters, a pitch pattern $\sigma_p(k)$ is generated in the frequency domain as illustrated in Fig. 19(b). The two spectral components $\sigma_f(k)$ and and $\sigma_p(k)$ are multiplied and normalized to yield the final spectral estimate for $\hat{\sigma}_{sR}(k)$,

$$\hat{\sigma}_{sR}(k) = \sigma_f(k)\, \sigma_p(k) \qquad k = 0, 1, 2, \cdots, M - 1. \quad (52)$$

This estimate, illustrated by Fig. 19(c), is then used for the bit assignment and step-size adaptation algorithms as seen in Fig. 18.

More generally, one may use other vocoder schemes, such as homomorphic vocoding to obtain a similar spectral fit. Although these schemes have not been tried, there are a number of factors that appear to lean in favor of the LPC model. From a theoretical point of view, the LPC model is closer to the physical mechanism of speech production than other models. In particular, the LPC model allows better spectral fits in the high Q formant regions where the signal must be encoded most accurately. From a practical point of view the use of the parcor parameters in the LPC model allows for a highly efficient means for quantizing the LPC coefficients. Since these techniques are well established [32], [33], they will not be discussed in this paper.

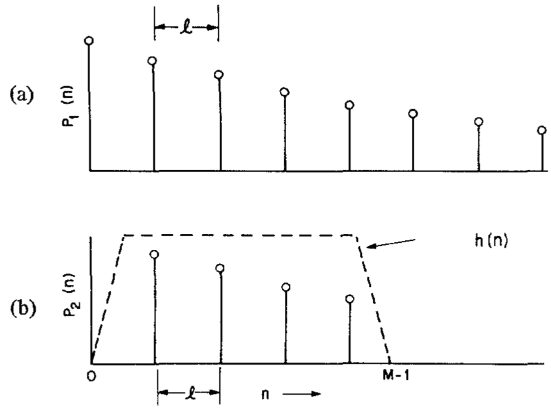


Fig. 20. (a) Infinite duration pitch model. (b) Windowed pitch model.

A number of alternatives are available for generating the pitch pattern in the frequency domain. We have investigated two different models. The first model is of the form

$$\sigma_p(\omega) = \left\| \frac{1}{1 - Ge^{-j\omega l}} \right\| \quad (53)$$

and is associated in time with a one-sided, infinitely long, periodic impulse train with exponentially decaying amplitudes, i.e.,

$$p(n) = \sum_{m=0}^{\infty} G^m \delta\, [n - ml]. \quad (54)$$

The model of (54) is depicted in Fig. 20(a). Because of the infinite duration of the assumed impulse train, this model leads to a very high Q model of the pitch harmonics in the frequency domain. As a consequence most of the bits are allocated to the pitch harmonics (at low bit rates), with essentially no transmission of DCT coefficients between these harmonics. This leads to a sensitivity of the algorithm to high-pitch speakers and to occasional pitch errors.

A slightly more realistic pitch model takes into account the fact that we are attempting to predict the spectral levels of a finite block of speech. In this model the infinitely long impulse train is windowed by the analysis filter $h(n)$, i.e.,

$$p(n) = h(n) \cdot \sum_{m=0}^{\infty} G^m \delta\, (n - ml) \quad (55)$$

as depicted in Fig. 20(b). In the frequency domain this amounts to the convolution of the frequency response of the impulse train, (53), with the frequency response of the window. Thus, the high Q pitch harmonics are effectively smoothed by the frequency response of the window which leads to a more realistic model.

Fig. 21 illustrates the operation of the "vocoder-driven" adaptation algorithm. Fig. 21(a) shows the DCT spectrum and the spectral estimate $\hat{\sigma}_{sR}(k)$ (seen as the dotted line). The speech block is the same as that of Fig. 17. Fig. 21(b) and 21(c) show the resulting bit allocation and decoded DCT spectrum in the receiver. The main effect of this algorithm is that it forces the assignment of bits to many pitch harmonics

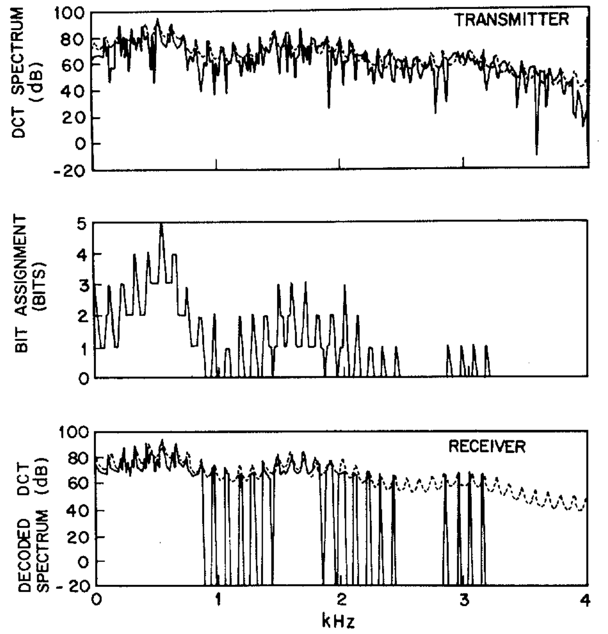


Fig. 21. Illustration of "speech specific" ATC algorithm.

which otherwise would not be transmitted at all, as seen by the comparison of Figs. 17 and 21. In addition, the algorithm helps to preserve the information in the pitch structure of the spectrum, even in frequency regions where one bit/sample is used (e.g., the region around 2 kHz in Figs. 17 and 21).

As seen by (40), the DCT coefficients can be expressed in terms of the magnitude of the Fourier transform $|U(k)|$ times a modulating term $\cos(\theta_k - \pi k/2M)$ where θ_k is the phase of the kth DFT coefficient. In principle, the magnitude term is predicted almost completely by the above algorithm and divided out, leaving only the cosine of the phase to be encoded.

The noise shaping for the vocoder-driven adaptation scheme should be based only on the smooth $(\sigma_f(k))$ component of the spectrum. Thus, for this algorithm the weighting $w_{sR}(k)$ of (51) is replaced by

$$w_{sR}(k) = \sigma_f^{2\gamma}(k) \qquad k = 0, 1, 2, \cdots, M-1. \quad (56)$$

In this way the noise shaping does not directly affect the allocation of bits in the pitch harmonics.

With the above "speech specific" algorithm the quality of the transform coder can be improved in the range of 16 to 8 kbits/s over that of the "nonspeech specific" algorithm. At 16 kbits/s and above both techniques have a similar quality. Below 16 kbits/s the nonspeech specific algorithm produces a low-level but highly discernible "burbling" or "click" noise which has been found to be quite annoying. This noise appears to be due primarily to a breakdown of the pitch structure and end effects in the blocks. With the speech specific algorithm a more parsimonious allocation of bits can be made which results in a significant reduction of this type of noise. As the bit rate of the coder is pushed down below 8 kbits/s, however, the algorithm becomes further starved for bits and these types of noises again become pronounced. In fact, at 4.8 kbits/s, the speech specific algorithm also produces significant degradations and click noise.

TABLE I
TYPICAL DESIGN PARAMETERS FOR THE "NONSPEECH SPECIFIC" ATC ALGORITHM AT 16 KBITS/S

Basic Parameters:	
Transform Size (M)	256
Sampling Rate (kHz)	8
Block Overlap (samples)	12
Max. No. Quantizer bits	5
Quantizer Loading (Q)	1.0
Noise Shaping Parameter (γ)	-0.125
No. of Side Info. Frequencies	20
Side Frequency Warping (λ)	
Voiced (λ_v)	-0.25
Unvoiced (λ_u)	0.25
No. of Bits for Quantization:	
Voiced/Unvoiced Decision	1
First Side Frequency	5
Remaining 19 Side Frequencies	38
Transform Coefficients	444
Total bits/block	488

G. Examples and Practical Considerations of Adaptive Transform Coder Designs

Computer simulations were generated for both of the above ATC algorithms corresponding, respectively, to Figs. 8 and 18. In addition, a number of modifications and refinements were made on the basic algorithms to enhance their performance and robustness. In this section we will briefly discuss a number of aspects of these designs in more detail. We will first consider issues that are common to both designs and then discuss the specifics of each design.

In both designs transform sizes of $M = 256$ (with a sampling rate of 8 kHz) were generally used. This size provides a sufficient spectral resolution to capture the fine details of the speech spectrum while keeping the overall delay of the coder within reasonable practical limits (less than 100 ms) for many types of communications applications.

Also, in both designs the bandwidth of the input speech was limited to the telephone bandwidth of 200 to 3200 Hz by an IIR digital filter prior to encoding. A similar bandpass characteristic was multiplied with the spectral estimate $\hat{\sigma}_{sR}(k)$ prior to the bit allocation and step-size adaptation algorithms. In this way all available bits are automatically constrained to be used within the 200 to 3200 speech band of interest and the performance of the coder is not affected by signals outside of this band.

Table I provides a summary of typical parameters that were used for the "nonspeech specific" ATC algorithm for a 16 kbits/s design. Blocks were overlapped by 12 samples and windowed by the trapezoidal window of Fig. 10. A maximum of 5 bits were allowed for quantizing each transform coefficient. A quantizer loading $Q = 1$ was used, and a noise shaping parameter of $\gamma = -0.125$ was found to give good results.

Twenty unequally spaced side frequencies were used in the coder with a different spacing used for voiced and unvoiced frequencies. The voiced/unvoiced decision was made according to a simple threshold decision on whether the signal energy was larger at low frequencies (near 500 Hz) or at high frequencies (near 2500 Hz). The choice of the unequal spacing of the side frequencies was determined from a set of equally spaced frequencies in the range 200 to 3200 Hz according to the relation [34]

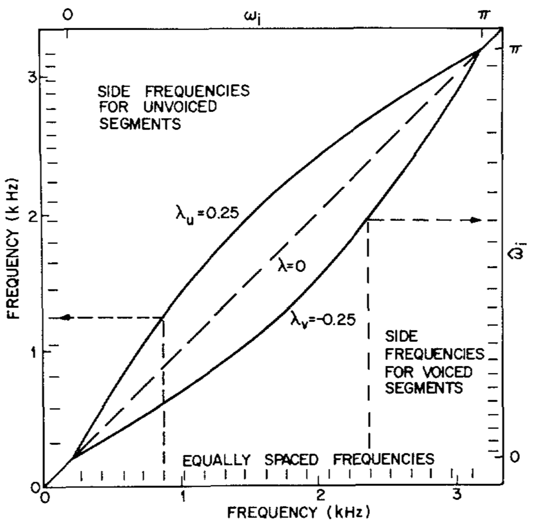


Fig. 22. Spectral warping used for unequal spacing of side information in "nonspeech specific" ATC example.

$$\hat{\omega}_i = \omega_i + 2A \tan \left[\frac{\lambda \sin \omega_i}{1 - \lambda \cos \omega_i} \right] \qquad i = 1, 2, \cdots 20$$

where ω_i (scaled to the range 200 to 3200 Hz) denotes the set of equally spaced frequencies $i = 1, 2, \cdots 20$ (expressed in radians) and $\hat{\omega}_i$ denotes the locations of the unequally spaced frequencies. The parameter λ is the warping parameter (*not* related to λ in Fig. 13) which determines the degree of warping of the unequally spaced frequencies. A value of $\lambda = \lambda_v = -0.25$ is used for the spacing of side frequencies when the speech energy is larger at low frequencies (voiced region). Similarly, a value of $\lambda = \lambda_v = 0.25$ is used for spacing of the side frequencies (unvoiced regions) is more predominant. Fig. 22 illustrates the location of these side frequencies for both the voiced and unvoiced cases. By choosing $\lambda = 0$ this scheme reduces to that of the equal spaced side frequencies depicted by Fig. 16.

Once the side frequencies are determined the local averages of the magnitude of the DCT values near those frequencies are computed. The logarithm of these values are then computed prior to quantization where a_i will be used to denote the logarithm of the ith side value. The value of a_i nearest to 500 Hz (for a voiced decision), or 2500 Hz (for an unvoiced decision), is quantized first with 5 bits of accuracy. The remaining coefficients are then quantized with 2 bits each by encoding the difference minus the expected difference from the ith to the $(i + 1)$th side value (where the expected difference is obtained from measurements of typical speech data). The step-size is also selected according to the expected variance of this difference. Thus, the scheme is similar to a DPCM coding of the a_i values starting from the one nearest to 500 or 2500 Hz and then quantizing differences in both directions from this starting value.

The above transform coding scheme (at 16 kbits/s) provides a quality that is essentially indistinguishable from an original 200-3200 Hz speech signal (based on our informal listening observations) over a wide range of speakers. It has a segmental S/N ratio on the order of 17 dB.

TABLE II
TYPICAL DESIGN PARAMETERS FOR THE "SPEECH SPECIFIC" ATC ALGORITHM AT 16, 12, AND 9.6 KBITS/S

Basic Parameters:	16 kb/s	12 kb/s	9.6 kb/s
Transform Size (M)	256	256	256
Sampling Rate (kHz)	8	8	8
Block Overlap (Samples)	8	16	16
Max. No. Quantizer bits	5	4	4
Quantizer Loading (Q)	1.0	1.3	1.5
Noise Shaping Paramter (γ)	-0.125	-0.125	-0.125
Order of LPC Analysis	12	12	12
No. Bits for Quantization:			
Gain	5	5	5
Pitch	5	5	5
Pitch Gain	4	4	4
Log Area Ratios:			
1	6	6	6
2	5	5	5
3	5	5	5
4	4	4	4
5	4	3	3
6	3	3	3
7	3	2	2
8	2	1	1
9	2	1	1
10	1	0	0
11	1	0	0
12	1	0	0
Data	445	316	244
Total Bits/Block	496	360	288

Table II provides a summary of typical parameters that were used for the "speech-specific" ATC algorithm at bit rates of 16, 12, and 9.6 kbits/s. As the bit rate was reduced the block overlap and quantizer loading parameters were increased to give a better subjective quality to the coder and, to some extent, to help reduce effects of "click" and "burbling" noises.

The side information was represented by a 12 pole LPC analysis. From this analysis, the log area ratios [32], [33] were computed and quantized. Prior to quantization the means of these values (obtained from a typical speech data) were subtracted. The quantization step-sizes were determined according to the expected values of the variances of the log area ratios (obtained from typical speech data). Table II shows the number of bits used to encode each log area ratio at the different bit rates of the coder.

The above "speech specific" ATC design can provide a quality that is essentially indistinguishable from the original 200-3200 Hz speech signal (based on our informal observation) at a bit rate of 16 kbits/s. The segmental S/N is on the order of 18 dB. At 12 kbits/s some slight degradations and occasional low-level clicks are observed for some speakers. Segmental S/N values on the order of 14.5 dB are observed at this bit rate. At 9.6 kbits/s a greater sensitivity to speakers is apparent. With some "good" speakers virtually no degradations or "clicks" are observable. However, with other speakers some distortion in the form of low level clicks or a slight hoarseness are noticeable but not overly disturbing compared to other coding schemes at this bit rate. A segmental S/N of about 12.8 dB is observed.

H. Discussion

In this section we have attempted to present a fairly detailed discussion of recent developments in adaptive transform coding. In addition, we have tried to provide an analysis/synthesis point of view of the transform coder which we believe will help in setting a framework for future research in this area.

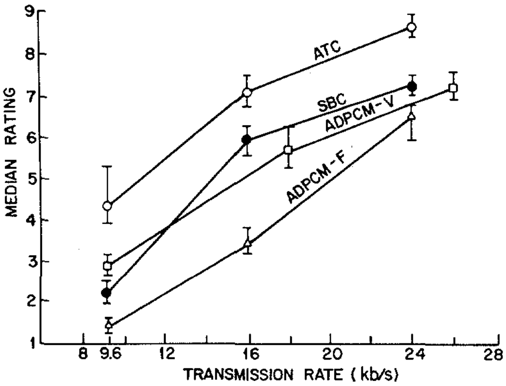


Fig. 23. Median opinion score ratings for comparison of coders.

V. Other Related Frequency Domain Coding Techniques and Modifications

In this paper we have primarily focused on sub-band coding and transform coding as two examples of frequency domain coders. These are not the only coders that belong in this class of coders however and in this section we wish to briefly mention other related techniques.

A. Phase Vocoder

Our discussion of frequency domain coders would not be complete without mention of the phase vocoder by Flanagan and Golden [6]. The phase vocoder is based on a direct implementation of the analysis/synthesis techniques discussed in Section II. In fact, the theory of short-time analysis/synthesis has been primarily developed through research on the phase vocoder.

In the phase vocoder the short-time spectral components $X_{sR}(k)$ are converted to magnitude and phase derivative components which are subsequently coded for transmission. Typically, 30 frequency channels are used in the phase vocoder which gives it a frequency resolution between that of the sub-band coder and the transform coder. Techniques for adaptively quantizing the channel signals of the phase vocoder, similar to those of sub-band and adaptive transform coding, can be used.

B. Polar Plane Coding

Another closely related technique is that of polar plane coding investigated by Gethoffer [35]. In this scheme the magnitude and phase of $X_{sR}(k)$ is computed and quantized with different accuracy. Good results were reported at bit rates below 16 kbits/s using very large (8192) transform sizes.

C. Voice-Excited and Vocoder-Excited Schemes

For very low bit rates (below 8 kbits/s), there are generally an insufficient number of bits to encode all of the significant frequency components. At these rates combinations of voice-excited vocoding and frequency domain coding techniques have been investigated by several researchers. Esteban *et al.* [36] have recently demonstrated that a combination of a sub-band coder and a voice-excited vocoder produce good results in the 9.6–4.8 kbits/s range. An interesting feature of their design is that they employ an LPC dynamic preemphasis (see Section II-E) to spectrally flatten the baseband signal prior to sub-band coding. In another approach Gold [37] has combined concepts of sub-band coding and channel vocoding for a multiple-rate speech coding/vocoding system.

VI. Conclusions

Except for the phase vocoder, most frequency domain coding techniques for speech have been proposed quite recently (within the past four years). In this paper we have attempted to draw together a general theoretical framework, based on analysis/synthesis and spectral estimation and modeling, which can be used as a foundation for further research in this direction.

Also, because of the recent origin of many of these techniques, little data is presently available on the comparison of the performance of frequency domain coding techniques with other waveform coding schemes. Preliminary studies, however, show that frequency domain coders can match and exceed the quality of their time domain counterparts.

Fig. 23 briefly summarizes the results of one such study [14]. In this study four different coders were compared at bit rates of 24, 16, and 9.6 kbits/s. The coders included a transform coder (ATC) based on the "nonspeech specific" algorithm (depicted by Figs. 8, 16, and 17), a sub-band coder (SBC), and two ADPCM (adaptive predictive PCM) coders, one with a fixed first-order predictor (ADPCM-F) and one with an 8th order adaptive predictor (ADPCM-V). All of the coders were nonpitch predicting coders (i.e., they did not exploit pitch prediction). Sixty-five listeners rated the coders in terms of quality on a 1 to 9 scale using 1 to represent the worst quality and 9 to represent the best quality. Quality at 1 was highly noisy and degraded, and quality at 9 was indistinguishable from the original. The median opinion scores of the listeners (bracketed by their 0.95 confidence interval) are plotted in Fig. 23 as a function of transmission rate. As seen, the ATC coder was clearly preferred over the other coders and the SBC coder was rated as having a quality comparable to that of the more complex ADPCM-V coder. A more detailed analysis of this data can be found in [14].

In another experiment involving a comparison of sub-band coding with ADPCM-F and two forms of delta modulation, a similar preference was found for the sub-band coder [38]. This was also substantiated in other informal comparisons found in [8] and [9].

With future developments of frequency domain coding techniques it is anticipated that even further improvements are possible with frequency domain techniques.

Acknowledgment

The authors are very grateful to the following persons for their comments and discussion during the course of this work: J. B. Allen, B. S. Atal, D. Falconer, J. L. Flanagan, P. Noll, and L. R. Rabiner. They would also especially like to thank P. Noll, R. Zelinski, and L. R. Rabiner for supplying Fortran code for a number of subroutines that were used in the computer simulations. Their assistance was very helpful and it is greatly appreciated.

References

[1] M. R. Portnoff, "Time scale modification of speech based on short-time Fourier analysis," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, Apr. 20, 1978.

[2] —, "Implementation of the digital phase vocoder using the fast Fourier transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 243–248, June 1976.

[3] J. B. Allen and L. R. Rabiner, "A unified approach to short-time Fourier analysis and synthesis," *Proc. IEEE*, vol. 65, pp. 1558–1564, Nov. 1977.

[4] R. W. Schafer and L. R. Rabiner, "Design and simulation of a speech analysis-synthesis system based on short-time Fourier analysis," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 165–174, June 1973.

[5] C. J. Weinstein, "Short-time Fourier analysis and its inverse," S. M. thesis, Elect. Eng. Dept., M.I.T., Cambridge, 1966.

[6] J. L. Flanagan and R. M. Golden, "Phase vocoder," *Bell Syst. Tech. J.*, vol. 45, pp. 1493–1509, Nov. 1966.

[7] J. B. Allen, "Short-term spectral analysis, synthesis, and modification by discrete Fourier transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-25, pp. 235–238, June 1977.

[8] R. E. Crochiere, S. A. Webber, and J. L. Flanagan, "Digital coding of speech in sub-bands," *Bell Syst. Tech. J.*, vol. 55, pp. 1069–1085, Oct. 1976.

[9] R. E. Crochiere, "On the design of sub-band coders for low bit-rate speech communications," *Bell Syst. Tech. J.*, vol. 56, pp. 747–770, May–June 1977.

[10] D. Esteban and C. Galand, "Application of quadrature mirror filters to split band voice coding schemes," in *Proc. 1977 Int. Conf. Acoust., Speech, Signal Processing*, Hartford, CT, May May 1977, pp. 191–195.

[11] R. Zelinski and P. Noll, "Adaptive transform coding of speech signals," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-25, pp. 299–309, Aug. 1977.

[12] J. D. Markel and A. H. Gray, Jr., *Linear Prediction of Speech.* New York: Springer-Verlag, 1976.

[13] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*. Englewood Cliffs, NJ: Prentice-Hall, 1978.

[14] J. M. Tribolet, P. Noll, B. J. McDermott, and R. E. Crochiere, "A comparison of the performance of four low bit rate speech waveform coders," *Bell Syst. Tech. J.*, vol. 58, pp. 699–712, Mar. 1979. Also, "A study of complexity and quality of speech waveform coders," in *1978 Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Apr. 10–12, 1978, Tulsa, OK, pp. 1586–1590.

[15] P. Cummiskey, N. S. Jayant, and J. L. Flanagan, "Adaptive quantization in differential PCM coding of speech," *Bell Syst. Tech. J.*, vol. 52, pp. 1105–1118, Sept. 1973.

[16] N. S. Jayant, "Digital coding of speech waveforms: PCM, DPCM, and DM quantizers," *Proc. IEEE*, vol. 62, pp. 611–632, May 1974.

[17] R. E. Crochiere, "A mid-rise/mid-tread quantizer switch for improved idle-channel performance in adaptive coders," *Bell Syst. Tech. J.*, vol. 57, pp. 2953–2955, Oct. 1978.

[18] D. J. Goodman and R. M. Wilkinson, "A robust adaptive quantizer," *IEEE Trans. Commun.*, pp. 1362–1365, Nov. 1975.

[19] J. M. Tribolet and R. E. Crochiere, "A vocoder-driven adaptation strategy for low bit-rate adaptive transform coding of speech," presented at 1978 Int. Conf. Digital Signal Processing, Florence, Italy, Sept. 1978.

[20] J. Huang and P. Schultheiss, "Block quantization of correlated gaussian random variables," *IEEE Trans. Commun. Syst.*, vol. CS-11, pp. 289–296, Sept. 1963.

[21] P. A. Wintz, "Transform picture coding," *Proc. IEEE*, vol. 60, pp. 809–820, 1972.

[22] H. P. Kramer and M. V. Mathews, "A linear coding for transmitting a set of correlated signals," *IRE Trans. Inform. Theory*, vol. IT-2, pp. 41–46, Sept. 1956.

[23] L. D. Davisson, "Rate-distortion theory and application," *Proc. IEEE*, vol. 60, pp. 800–808, July 1972.

[24] N. Ahmed and K. R. Rao, *Orthogonal Transforms for Digital Signal Processing*. New York: Springer-Verlag, 1975.

[25] W. Chen and S. C. Fralick, "Image enhancement using cosine transform filtering," in *Proc. Symp. Current Mathematical Problems in Image Science;*" Montery, CA, Nov. 1976, pp. 186–192.

[26] —, "A fast computational algorithm for the discrete cosine transform," *IEEE Trans. Commun.*, vol. COM-25, pp. 1004–1009, Sept. 1977.

[27] M. J. Narasimha and A. M. Peterson, "On the computation of the discrete cosine transform," *IEEE Trans. Commun.*, vol. COM-16, pp. 934–936, June 1978.

[28] J. Max, "Quantizing for minimum distortion," *IRE Trans. Inform. Theory*, vol. IT-6, pp. 7–12, Mar. 1960.

[29] R. E. Crochiere, L. R. Rabiner, N. S. Jayant, and J. M. Tribolet, "A study of objective measures for speech waveform coders," in *1978 Proc. Zurich Seminar on Digital Commun.*, Zurich, Switzerland, Mar. 1978, pp. H1.1–H1.7.

[30] B. S. Atal and M. R. Schroeder, "Predictive coding of speech signals and subjective error criteria," in *1978 Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Tulsa, OK, Apr. 10–12, 1978, pp. 573–576.

[31] M. Berouti and J. Makhoul, "High quality adaptive predictive coding of speech," in *1978 Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 1978, Tulsa, OK, Apr. 10–12, 1978, pp. 303–306.

[32] A. H. Gray, Jr. and J. D. Markel, "Quantization and bit allocation in speech processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 459–473, Dec. 1976.

[33] R. Viswanathan and J. Markhoul, "Quantization properties of transmission parameters in linear predictive systems," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 309–321, June 1975.

[34] A. V. Oppenheim and D. H. Johnson, "Discrete representation of signals," *Proc. IEEE*, vol. 60, pp. 681–691, June 1972.

[35] H. Gethoffer, "Polar plane block quantization of speech signals using bit-pattern matching techniques," in *1977 Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Hartford, CT, May 9–11, 1977, pp. 200–203.

[36] D. Esteban, C. Galand, D. Mauduit, and J. Menez, "9.6/7.2 kbits/s voice excited predictive coder (VEPC)," in *1978 Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tulsa, OK, Apr. 10–12, 1978, pp. 307–311.

[37] B. Gold, "Variable rate speech processing," presented at the 1978 Int. Conf. Digital Signal Processing, Florence, Italy, Sept. 1978.

[38] D. J. Goodman, C. Scagliola, R. E. Crochiere, L. R. Rabiner, and J. Goodman, "Objective and subjective performance of tandem connections of waveform coders with an LPC vocoder," *Bell Syst. Tech. J.*, vol. 58, Mar. 1979, pp. 601–629.

[39] R. Zelinski and P. Noll, "Adaptive block quantization of speech signals," (in German) Heinrich-Hertz-Institute, Berlin, Germany, Tech. Rep. 181, 1975.

[40] —, "Approaches to adaptive transform speech coding at low bit rates," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 89–95, Feb. 1979.

[41] J. L. Flanagan, *Speech Analysis Synthesis and Perception*. New York: Springer-Verlag, 1972.

[42] R. E. Crochiere, "A novel approach for implementing pitch prediction in sub-band coding," in *1979 Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Washington, DC, Apr. 2–4, 1979, pp. 526–529.

[43] A. J. Barabell and R. E. Crochiere "Sub-band coder design incorporating quadrature filters and pitch prediction," in *1979 Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Washington, DC, Apr. 2–4, 1979, pp. 530–533.

[44] J. L. Flanagan *et al.*, "Speech coding," *IEEE Trans. Commun.*, vol. COM-27, pp. 710–737, Apr. 1979.